Inferring Complex Activities for Context-aware Systems within Smart Environments

April 2020

Darpan Triboan (MSc, BSc Hons.)

Thesis submitted for the degree of Doctor of Philosophy

DECLARATION

I declare that the work was solely conducted during registration for the above award with De Montfort University, under University supervision.

I declare that no material contained in the thesis has been used in any other submission for an academic award.

I confirm that the work represented in this submission was undertaken solely by myself except where otherwise attributed.

I certify that I have not used another person to proofread, edit or amend my work, unless acting in strict compliance with the regulations outlined in the Code of Practice. In the event that this statement is proved to be false I acknowledge that I will have committed an academic offence.

I acknowledge the copyright and other intellectual property rights in relation to my thesis and other work prepared and submitted in the course of my studies shall belong to the University, except where specifically agreed otherwise by the University in writing.

ACKNOWLEDGEMENTS

I would like to start by thanking Almighty Lord Krishna for giving me the opportunity, perseverance, and wisdom to undertake this project.

During the project, I received help from many people all of whom I would like to thank. Though, it is not possible to thank everyone separately, I would like to express my gratitude to a few distinct individuals. First of all, I would like to thank my supervisors, Professor Liming Chen, Dr Feng Chen, Dr Francois Siewe and Professor Zumin Wang, whose suggestions, critiques, and continuous support and encouragements made my PhD journey exciting and rewarding. More specifically, I would like to express my sincere gratitude and admiration to Prof. Liming Chen for sharing his valuable research expertise, his enthusiasm, constant availability during and outside working hours and creating endless opportunities for me to learn and excel in both, professional and personal development. I also wish to thank Dr Feng Chen for his feedback/comments, suggestions, and inquisitive questions that always led me to reflect on the chosen approaches. My appreciation also goes to Dr Francois Siewe who introduced me to the whole paradigm of ubiquitous computing during the post-graduate course at De Montfort University (DMU) and guidance throughout the PhD journey.

Secondly, I am heartiest thanks to my family – mum, dad, sister, and brother-in-law – who showered their love, care, and compassion when it mattered regardless of my absence. My deepest and most heartfelt gratitude goes to Anisha Meggi who encouraged me to peruse this Doctoral Research and continues to support and inspire me every day.

Thirdly, I would like to thank the following people who I had special opportunity to learn, share and get inspired from the random conversations throughout the PhD journey: Ismini Psychoula, Sarah Fallmann, Khawla Al Hasan, Ruijie Wang, Yumei Zheng, and Suleiman Aliyu. Furthermore, my warm gratitude goes to Thomas Blondelle and Simon Forest from the University of Lyon who contributed to the development and evaluation of the smart lab system deployed in CIIRG.

Fourthly, I would like to thank all the staff at the Cyber Technology Institute (CTI)/ Software Technology Research Laboratory (STRL), and the Doctoral College, who in one way or another made this project feasible. Moreover, I wish to recognise the role of DMU that made it possible for me to undertake this project through the DMU Research Student Scholarship.

Lastly, my sincere appreciation to my examination panel: Prof. Lu Liu (external examiner), Dr George Okeyo (internal examiner) and Prof. Ali Hilal Al-Bayatti (chair) for their critiques and valuable comments during the PhD Viva Voce.

ABSTRACT

The rising ageing population worldwide and the prevalence of age-related conditions such as physical fragility, mental impairments and chronic diseases have significantly impacted the quality of life and caused a shortage of health and care services. Over-stretched healthcare providers are leading to a paradigm shift in public healthcare provisioning. Thus, Ambient Assisted Living (AAL) using Smart Homes (SH) technologies has been rigorously investigated to help address the aforementioned problems.

Human Activity Recognition (HAR) is a critical component in AAL systems which enables applications such as just-in-time assistance, behaviour analysis, anomalies detection and emergency notifications. This thesis is aimed at investigating challenges faced in accurately recognising Activities of Daily Living (ADLs) performed by single or multiple inhabitants within smart environments. Specifically, this thesis explores five complementary research challenges in HAR. The first study contributes to knowledge by developing a semantic-enabled data segmentation approach with user-preferences. The second study takes the segmented set of sensor data to investigate and recognise human ADLs at multi-granular action level; coarse- and fine-grained action level. At the coarse-grained actions level, semantic relationships between the sensor, object and ADLs are deduced, whereas, at fine-grained action level, object usage at the satisfactory threshold with the evidence fused from multimodal sensor data is leveraged to verify the intended actions. Moreover, due to imprecise/vague interpretations of multimodal sensors and data fusion challenges, fuzzy set theory and fuzzy web ontology language (fuzzy-OWL) are leveraged. The third study focuses on incorporating uncertainties caused in HAR due to factors such as technological failure, object malfunction, and human errors. Hence, existing studies uncertainty theories and approaches are analysed and based on the findings, probabilistic ontology (PR-OWL) based HAR approach is proposed. The fourth study extends the first three studies to distinguish activities conducted by more than one inhabitant in a shared smart environment with the use of discriminative sensor-based techniques and time-series pattern analysis. The final study investigates in a suitable system architecture with a real-time smart environment tailored to AAL system and proposes microservices architecture with sensor-based off-the-shelf and bespoke sensing methods.

The initial semantic-enabled data segmentation study was evaluated with 100% and 97.8% accuracy to segment sensor events under single and mixed activities scenarios. However, the average classification time taken to segment each sensor events have suffered from 3971ms and 62183ms for single and mixed activities scenarios, respectively. The second study to detect fine-grained-level user actions was evaluated with 30 and 153 fuzzy rules to detect two fine-grained movements with a pre-collected dataset from the real-time smart environment. The result of the second study indicate good average accuracy of 83.33% and 100% but with the high average duration of 24648ms and 105318ms, and posing further challenges for the scalability of fusion rule creations. The third study was evaluated by incorporating PR-OWL ontology with ADL ontologies and Semantic-Sensor-Network (SSN) ontology to define four types of uncertainties presented in the kitchen-based activity. The fourth study illustrated a case study to extended single-user AR to multi-user AR by combining RFID tags and fingerprint sensors discriminative sensors to identify and associate user actions with the aid of time-series analysis. The last study responds to the computations and performance requirements for the four studies by analysing and proposing microservices-based system architecture for AAL system. A future research investigation towards adopting fog/edge computing paradigms from cloud computing is discussed for higher availability, reduced network traffic/energy, cost, and creating a decentralised system.

As a result of the five studies, this thesis develops a knowledge-driven framework to estimate and recognise multi-user activities at fine-grained level user actions. This framework integrates three complementary ontologies to conceptualise factual, fuzzy and uncertainties in the environment/ADLs, time-series analysis and discriminative sensing environment. Moreover, a distributed software architecture, multimodal sensor-based hardware prototypes, and other supportive utility tools such as simulator and synthetic ADL data generator for the experimentation were developed to support the evaluation of the proposed approaches. The distributed system is platform-independent and currently supported by an Android mobile application and web-browser based client interfaces for retrieving information such as live sensor events and HAR results.

PUBLICATIONS

Journals

1. D. Triboan, A. Meggi, "Reformulating a Smart Home System for the Indian Context: Diu Island", International Journal of Design & Nature and Ecodynamics, vol. 14, no. 4, pp. 299-310, 2019.
2. D. Triboan, L. Chen, F. Chen, "A Semantics-based Approach to Sensor Data Segmentation in Real-time Activity Recognition", Special Issue on "Smart Cyber-Physical Systems: towards Pervasive Intelligence systems", Future Generation Computer Systems, vol. 93, 2019, pp. 224-236.
3. D. Triboan, L. Chen, F. Chen, Z. Wang, "Semantic Segmentation of Real-time Sensor Data Stream for Complex Activity Recognition", Pervasive and Ubiquitous Computing, special issue, 2016.
4. D. Triboan, L. Chen, F. Chen, Z. Wang, "Towards Service-oriented Architecture for a Mobile Assistive System with Real-time Environmental Sensing", Tsinghua Science And Technology, vol. 18, no.3, 2016.

Conferences

5. D. Triboan, L. Chen, F. Chen, "Fuzzy-based Object-usage Level Human Activity Recognition within Smart Environment", 2019 IEEE Smart World Congress, Leicester, UK, 2019. (***Best Paper Awarded***)
6. D. Triboan, L. Chen, F. Chen, S. Fallmann, I. Psychoula, "Real-Time Sensor Observation Segmentation For Complex Activity Recognition Within Smart Environments", 2017 IEEE Ubiquitous Intelligence and Computing (UIC), San Francisco, CA, USA, August 4-8, 2017.
7. S. Fallmann, I. Psychoula, L. Chen, F. Chen, J. Doyley, D. Triboan, "Reality and Perception: Activity monitoring and data collection within a real-world smart home", 2017 IEEE Ubiquitous Intelligence and Computing (UIC), San Francisco, CA, USA, August 4-8, 2017.
8. D. Triboan, L. Chen, F. Chen, "Towards a Mobile Assistive System Using Service-oriented Architecture", 2016 IEEE Symposium on Service-Oriented System Engineering (SOSE), Oxford, UK, 2016, pp. 215-224.

Book Chapters

9. D. Triboan, L. Chen, F. Chen, and Zumin Wang, "Multigranularity Activity Recognition within a Multiple Occupancy Environment", Cyber-Enabled Intelligence, CRC Press, 2019, pp.149-167.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AR	Activity Recognition
AAL	Ambient Assisted Living
ADL	Activities of Daily Living
AJAX	Asynchronous JavaScript and XML
A-Box	Assertion box
API	Application programming interface
BLE	Bluetooth
BN	Bayesian Network
CRUD	Create, Read, Update and Delete
CEP	Complex Event Processing
DD	Data-driven
DL	Description Logic
DST	Dempster-Shafer Theory
ESB	Enterprise Service Bus
EC	Event Calculus
HCI	Human-computer Interface
HAR	Human Activity Recognition
HTTP	Hypertext Transfer Protocol (HTTP)
HTML	Hypertext Transfer Protocol
HMM	Hidden Markov Model
ICT	Information and Communications Technology
IOT	Internet of Things
OS	Operating System
JESS	Java Expert System Shell
JSON	JavaScript Object Notation
KD	Knowledge-driven
MEBN	Multi-entity Bayesian Network
MSA	Microservice Architecture
MQTT	Message Queuing Telemetry Transport
OWL	Web Ontology Language
PR-OWL	Probabilistic Ontology
RDBMS	Relational Database Management System
REST	Representational State Transfer
RDF	Resource Description Framework
RDFS	RDF Schema
RF	Radio Frequency
SE	Smart Environment
SOA	Service-oriented Architecture
SOAP	Simple Object Access Protocol
SH	Smart Home
SPARQL	SPARQL Protocol and RDF Query Language
SSN	Semantic Sensor Network
SVM	Support Vector Machine
SWRL	Semantic Web Rule Language
T-Box	Terminology box
TDB	Triplestore Database
USB	Universal Serial Bus
WSN	Wireless Sensor Network
WWW	World Wide Web
W3C	World Wide Web Consortium
XML	Extensible Markup Language

CHAPTER 1. INTRODUCTION

The global ageing population has been projected to reach over 2 billion by 2050 [1], [2]. The enhancement in human lifespan has led to new challenges and adverse effects. One of the major concerns is that increasing age-related diseases has created a greater demand for health and social care services to provide high-quality care with limited resources [3].

To address this problem, many academics, health service providers and corporations have explored the opportunities created by the rapid advancement of recent state-of-the-art technologies [4] which attempt to imitate some of the health care professional's services in the comfort of the user's own home. As technology becomes more ubiquitous, it is now possible to monitor human physiology, metaphysical and physical behaviours through diverse technologies within smart homes (SH). With the knowledge obtained from the inhabitant's context, intentions and past action data gathered from sensors, a system can be trained to recognise human actions, learn, adapt, automate daily tasks to provide timely assistance. Consequently, Ambient Assisted Living (AAL) systems [5]–[7] are being developed that make use of the cutting-edge SH technologies to collect data, perform activity recognition (AR) and provide real-time context-aware assistance to inhabitants. The goal of AAL systems is to empower the elderly for independent living and assist health care professionals in providing effective and timely health and care services.

At present, researchers are rigorously investigating various aspects of building context-aware AAL systems. Human Activity Recognition (HAR) plays a critical role in the AAL system to analyse inhabitant's actions and provide timely assistance when required. Hence, this thesis specifically focuses on the investigation of recognising a simple or mixed activities, i.e., interweaved and concurrent in the context of single and multiple users (interchangeably referred to as inhabitant). Researchers have investigated heavily in recognising a single activity at a given time and commonly performing sub-tasks sequentially. However, there has been little investigation to recognise complex activities such as interleaving and concurrent activities (also referred to as a composite/mixed activity) performed by a single or multiple users. Besides, several challenges and opportunities are required to be explored at all the levels of activity recognition (AR) phases, from activity modelling and representing to provisioning assistance.

1.1. Research Background

HAR has long been studied using image and video analysis technologies in computer vision. Different from the vision-based approach, this thesis addresses AR problems by inferring an inhabitant's behaviour from a series of observations of sensors monitoring the inhabitant's actions [8]. AR involves five main tasks: (a) monitoring of user interactions with objects, (b) sensor data (pre)processing, (c) activity modelling, (d) inferring activities from sensor data against activity models, and (e) activity learning. CHAPTER 2 elaborates on each of these tasks which can be undertaken in different ways, and each has its specific challenges, making AR a multidimensional research problem. However, the focus of the research problems and issues related to tasks (a)-(d) being investigated in this thesis are highlighted in section 1.2. AR has recently attracted increasing attention as it plays a vital role in the emerging new wave of applications which support context awareness, multimodal interactions, personalisation and adaptation, e.g. AAL in SH.

An SH or a smart environment (SE) is an augmented living environment whereby the sensors, actuators and computer systems are interconnected and deployed within the inhabitant's environment to monitor their behaviour and provide assistance as and when needed. Recognising ADLs in SHs is critical to enable techniques for advanced AAL features, e.g., the provision of timely context-aware assistance, the discovery of behaviour patterns for personalisation, and the detection of behaviour changes for risk prediction, prevention, and adaptive healthcare. Though AR in the context of SH based AAL has been studied over the past decade, the techniques, however, are still far from mature.

1.2. Research Problems and Issues

This thesis seeks to investigate four main challenging aspects of building an AAL system. These challenges are to (1) cultivate a suitable system architecture using open source and off-the-shelf products, (2) develop a segmentation approach on continuous sensor observations from the heterogeneous sensor network, (3) handle imprecise and uncertainty factors within SH environments for single and (4) multi-user AR approach at multi-level of granularity.

The first challenge is to develop an AAL system architecture with consideration of several non-/functional system requirements, hardware and software availabilities. The common requirements for the AAL system architecture are to be open source, reusable, expandable, interoperable and scalable. The open-source nature of the system enables wider communities to engage with the development of the system, be creative with the system's functionalities, and integrate third-party resources to enrich the system's capabilities further. In contrast, the proprietary software or hardware components incur licencing costs, restrict one into following the company's policy and trusting the companies to provide law-abiding services. As such, it poses constraints for the system to be adaptable by the wider community, difficult to reuse and interoperate the components with other manufacturer's resources; ultimately slowing down the expansion process. On the other hand, many Internet-of-Things (IoT) enabled off-the-shelf hardware and software components are open-source. The diversity of sensing technologies, communication protocols adopted by manufactures with little documentations on third-party device integration are some of the barriers faced in developing a SH solution.

The second challenge is to pre-process the continuous sensor data stream from a given SH environment and segment the data relevant to the ongoing ADLs. Existing studies adopt ADL information model or fixed/dynamic time windowing mechanisms to segment incoming sensor events. These studies achieved respective success in detangling ADLs conducted by Human being in a wide range of manners, i.e. sequentially, interleaved or concurrently. The limitations of existing approaches are that sensor data are stored into the database first and then retrieved to reason with the sensor, which requires constant read/write operations with the database and scripting bespoke queries. Besides, these studies mainly focus on generic ADLs, but in a real-world setting, users perform ADLs based on their preferences, cultural rituals or for medical reasons. Therefore, the challenge for the segmentation approach is to include personalised action/sensor observations to support the subsequent task of AR to reason with the data accurately. This should allow the user to have more control over the system by enabling personalisation and adapting capabilities. Although the static generic ADL modelling and personalisation can be strongly supported by learning algorithms, more work is still required to evaluate the confidence level, i.e., allowing the user to verify and validate the inferencing results as per their evolving needs.

The third challenge is to perform accurate single-user AR for the ADLs performed in a SE. The main problem for recognising ADLs is that there are many ADLs, and each ADL can be conducted in diverse ways. Different from traditional pattern recognition, which is based on off-line static datasets, AAL requires that AR be performed dynamically and continuously in real-time, thus allowing just-in-time context-aware assistance. Current AR research has focused on two well-defined simplified activity scenarios: sequential activities where ADLs are conducted one after another; and mixed activities where several activities are performed simultaneously, either interleaved or concurrently. Furthermore, testing and evaluations of these AR studies have been based on experiments of a scripted or pre-segmented sequence of sensor observations. These ideal scenarios do not reflect the way ADLs are performed in real-world living environments within which sequential, interleaved and concurrent activities are often mixed in a variety of permutations. Nevertheless, little effort has so far been made to address complex issues in recognising non-/sequential, interleaved, and concurrent activities (referred to as mixed activities hereafter). Moreover, several uncertainty factors (i.e., sensing failure/missing sensor, mishaps and forgetfulness) and fuzzy/imprecise non-binary sensor data interpretation can pose severe challenges in estimating successful completion of actions in a given ADL. This thesis aims to bridge this knowledge gap by developing a semantics-enabled generic approach to inferring mixed activities and estimating confidence level based on emergent behavioural semantics from streaming sensor data (interactions with objects).

Finally, while most of the research in AAL systems focus on single-user AR, fewer studies have tackled multiple-user AR within a shared SE. In a real-world setting, more than one user is likely to share the same space in a given time and detecting how many and who is in the environment is essential for two main reasons. The first reason is to identify who is interacting within SE is to personalised AR recognition algorithms based on individual user's medical needs and personal preferences. For example, Bob may have diabetes and will add sweetener, while Alice may prefer to add sugar and ginger when making a tea within a shared space and time. Therefore, the challenge of associating the individual actions with a set of everyday objects to the user becomes important to prompt/remind the user that they have missed an essential action. The second reason is that multiple users may collaboratively or independently perform a single or mixed activity which makes it difficult to distinguish the individual and provide personalised assistance.

1.3. Aims and Objectives

The overall aim of this research project is to accurately recognise sequential and mixed activities conducted by single and multi-user within shared SH environments to support AAL applications. The focus of the research will be on the ADL knowledge modelling and reasoning methodology to incorporate imprecise and uncertainty factors influencing the mixed activities recognition results within SH environment. Nevertheless, some research contribution will also be made to identify and integrate some off-the-shelf and bespoke SH devices adequately in the overall system architecture. The proposed AR approach will be tested and evaluated at various stages of the project, and all the novel findings will be published within the research community in the appropriate forms, i.e. conference/journal papers and thesis.

The key objectives are

1. To conceptualise and develop an activity model which can be semantically and formally processed when inferring and recognising mixed user activities.
2. To conceive, develop and evaluate a semantic-enabled algorithm to disentangle the continuous sensor observations into relevant ADLs from the SH environment in real-time.
3. To conceive, develop and evaluate an AR algorithm which recognises single-user activity at the multi-granularity action level.
4. To formulate and develop a framework to incorporate factors of uncertainties within a single-user mixed AR algorithm.

5. To conceive and develop a multi-user activity recognition approach with unobtrusive sensing environment.
6. To investigate and develop a system architecture after investigating into the existing AAL systems and SH technologies to build an open-source, interoperable, reusable and expandable system.
7. To build, deploy and evaluate the system implementation using syntactic/real datasets containing varying use case scenarios.
8. To publish the findings and contributing to the wider community.

1.4. Methodology and Scope

The fundamental difference between the research of this thesis and previous AR approaches is that semantics of individual sensor observations, and the emergent semantics of aggregated sensor observations are analysed for real-time dynamic processing. Here the semantics of a sensor observation refers to the potential function of an object to which the sensor is attached, and the emergent semantics of several sensor observations refers to the joint semantics of these individual sensor observations, i.e., the potential function these objects can realise. For example, the semantics of the sensor activation attached to a *cup* is "*drink container*". The emergent semantics of the sensor observations for a *cup*, a *teabag*, *hot water* and *milk* is "*preparing tea*". Emergent semantics could refer to an ADL or part of an ADL.

The hypothesis is that if real-time streaming sensor observations can be dynamically separated to multiple sequences with each corresponding to an emerging behaviour, i.e. one ADL, then activity recognition (AR) can be achieved through simple AR. There is no need for complex or mixed activity models as they emerge from the aggregated semantics of individual actions. Simple activity in this thesis is referred to ADLs defined by[9], whereby, each activity serves one purpose only, and their occurrences are independent of each other. This hypothesis essentially breaks down the problem of mixed activity recognition into several simpler issues, namely the dynamic separation and segmentation of sensor observations, simple activity modelling and recognition, and the learning of inter-/intra-activity temporal dynamics to characterise a mixed activity. The resulting techniques will be capable of discerning and aggregating real-time sensor observations on-the-fly, dynamically recognising simple and then inferring the mixed activities, learning inter-/intra- action relationships and activity patterns for behaviour analysis. Ultimately, semantic-based sensor segmentation provides a robust systematic solution for mixed AR, which applies to a wide range of real-world use scenarios.

Sequential and mixed AR has been studied separately in the past. A well-established approach often dubbed as the data-driven (DD) approach which uses data mining and machine learning techniques to construct activity models from pre-existed datasets and then use the models as classifiers to map sensor data streams to corresponding activity labels. The approach includes generative methods, e.g. Markov models [10] and Bayesian networks [11]–[13], and discriminative methods, e.g., support vector machine [14] and decision tree[15], [16]. An alternative approach is to exploit rich domain knowledge for activity modelling and formal logical reasoning for activity recognition, which is usually referred to as the knowledge-driven (KD) approach due to its use of knowledge engineering techniques. Related work includes web mining [17] and formal logical modelling and inference [18]. Both approaches have their advantages and drawbacks. For instance, the cold-start and model reusability problems for the DD approach, and in contrast, the model rigidity and the inability to handle uncertainty for the KD approach. To address DD and KD approaches shortfalls, transfer learning [19] and an ontology-based hybrid approach [20] have been explored. Nevertheless, these efforts have mainly focused on improving sequential activity recognition.

Both DD and KD approaches have been investigated for sequential and mixed AR, e.g. Factorial Conditional Random Field (CRF) [21], Skip-chain CRF[22], hierarchical HMM model [23], context-driven activity theory, the ontology-based semantic reasoning[24], [25]**,**. These studies assume that there exist fixed action patterns within the constituent activities of a mixed activity, and between them. As such, they create a single model for each mixed activity either in advance using knowledge engineering techniques or later using data mining methods. However, this assumption is too strong as in real life as each ADL can be performed in different ways, and multiple ADLs can be interleaved and performed concurrently in many permutations. It is difficult to specify all mixed activity models in advance or to obtain large datasets to construct a complete model by training. As a result, these approaches are difficult to apply to real-world use scenarios.

Mixed activity recognition poses more challenging research issues than sequential activity recognition due to the random and spontaneous nature ADLs are performed. A DD pattern mining approach [26] has been studied for this purpose, but it still inherited the drawbacks of learning-based methods. In general, limited work has been done in mixed activity recognition which is demanding a vigorous investigation to meet the requirement of behaviour analysis of real-world applications. Solving these inherited and unique problems of mixed activity recognition is one of the focus areas of this project.

The development of DISENTANGLE approach is separated into four phases, as shown in Figure 1.1. In Phase 1, each observation from streaming sensor data is assessed to decide if the observation represents the start of a new activity or the continuation of ongoing activity. The decision is made based on the semantic relevance and compatibility of this sensor observation with the preceding observations. For example, the consecutive actions, "*turning on cooker*" and "*picking up a smartphone*" will indicate two distinct activities in terms of the observation semantics, while "*turning on cooker*" and "*picking up a pan*" will most likely be associated to one activity. Phase 1 separates the sensor data into multiple data sequences with each of them corresponding to a simple ADL. However, semantical-based data segmentation mechanisms alone will not be sufficient to handle the complexity of sensor data of mixed activities. Future work shall use dynamic time windows [27] and emergent semantics to support continuous data segmentation from the SH environment. Phase 2 is responsible for analysing segmentation data to perform simple/mixed activity recognition using the semantical description of ADLs. Phase 2 also perform imprecise data and uncertainty reasoning on the continuous sensor data stream. The intermediate results of simple activity recognition are used by the semantic segmentation method to extract behaviour semantics of the ongoing activity for segmentation purpose.

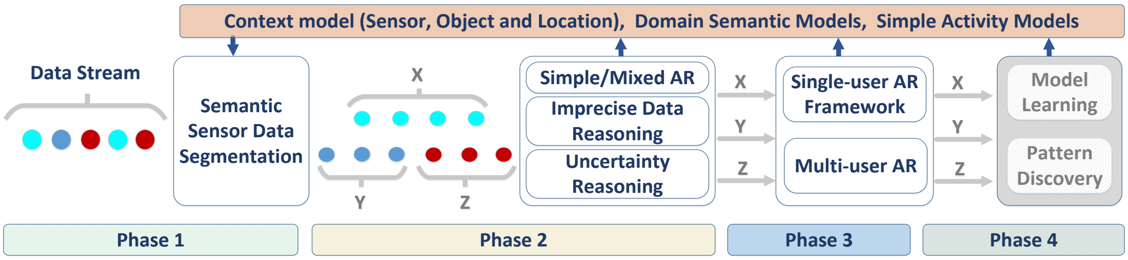


Figure 1.1. The semantics-enabled generic approach to mixed activity recognition

In parallel with simple activity recognition in Phase 2, the inter-activity relations, i.e. temporal and sequential information between these emerging simple activities are analysed online, which leads to mixed activity recognition. Moreover, user's actions with the environment are interpreted by fusing multimodal sensor data to accurately determine completion of fine-grained level actions, i.e., "*pouring*" hot water from a kettle or "*drinking*" from a cup. In Phase 3, single-user activity recognition framework is developed to incorporate methods developed in Phase 2 for recognising incomplete and missing user actions. Furthermore, detecting, identifying and associating multi-user actions within a shared living environment is explored. In Phase 4, the activity traces of the recognised simple and mixed activities from Phase 2 and 3 are analysed predominately offline to discover new ADLs and user-specific action patterns to enhance the initial models. This will subsequently improve semantic segmentation and recognition through iterative model adaptation. Although Phase 4 is not the focus of the thesis, the notion of activity learning for a complete system or adapting to user's needs are taken into consideration throughout the thesis and will be the foundation of future work.

Developing models and methods to support tasks in each phase is the major work of this thesis. While typical research methods are followed for each specific research issue, i.e.

literature review, gap analysis, technique development and evaluation, special attention is paid to technique and system integration between tasks and phases. To this end, ontologies and semantic reasoning are used as the unified conceptual backbone for domain modelling, representation and inference. In particular, ontological modelling is extended and/or expanded to enhance their capabilities for handling temporal relationships, fuzzy/ambiguous concepts and uncertainty. In future work, novel learning methods which can consume semantically enriched sensor data and ADL traces will be developed for learning new activities, improving models and discovering behaviour patterns.

Techniques developed in study is evaluated using a quantitative research method using a) lab-based smart environments with simulated real-world use scenarios/case studies, and b) using real user ADL datasets and syntactical data. Their performances are evaluated in three activity scenarios, i.e. sequential activities, mixed activities and error-prone/missing action in mixed activities to facilitate comparison with existent research which uses the same datasets.

1.5. Main Contributions

The key contributions of this study are as follows:

1. A multi-layered ADL model containing fuzzy and probabilistic knowledge to recognise single and multi-user activities at fine-grained action level within the SH domain. In addition, the model reuses external vocabularies to expand the knowledge model with domain-specific information such as Semantic Sensor Network (SSN).
2. A semantic-enabled real-time sensor data segmentation algorithm to disentangle mixed activities with user preferences in a smart environment.
3. A single-user fine-grained level activity modelling and recognition approach with fuzzy sensor observations and data fusion.
4. A probabilistic reasoning method to incorporate factors of uncertainties within the AR algorithm.
5. A knowledge-driven framework to handle imprecise knowledge and uncertainty factors when estimating ADL.
6. A multi-user activity recognition approach with discriminative sensors and time-series analysis.
7. A microservice-based system architecture tailored for AAL and SH equipped with multimodal sensing approaches.

1.6. Outline of the Thesis

The remainder of the thesis is organised as follows chapters. *Chapter 2* presents recent studies in the realm of human activity recognition (HAR). It initially introduces the HAR, the role it plays in ambient assisted living (AAL) system and the challenges of developing the real-time AAL system. It then examines and reviews critical studies for the three critical components of HAR (activity modelling, data collection, and processing), and the selection of AAL system architecture. Subsequently, the chapter presents a summary of the issues and challenges in the development of HAR enabled AAL systems.

Chapter 3 develops semantics-enabled methods for dynamic separation and segmentation of streaming sensor observations (a) into different threads of data relevant to ADLs, and (b) incorporate user preferences during this phase. This chapter first reviews key studies on continuous online segmentation of the sensor data streams and highlight the issues being investigated. Next semantical segmentation approach is presented which incorporate user preferences in knowledge modelling, semantic decision engine and segmentation algorithm. Subsequently, implementation details and evaluation results on the accuracy and performance of the proposed semantic segmentation is presented. This chapter then summarises the findings with the proposed semantic-enabled sensor data segmentation and discusses the direction for the future.

Chapter 4 focuses on developing AR approach at the multi-granularity level on a given segmented set of sensors relevant to ADLs. This chapter initially describes the motivation for analysing sensor observation at a coarse-/fine-grained level and the challenges of reasoning with imprecise sensor measurements and fusioning multimodal sensor data. Subsequently, the key studies detecting ADLs at multi-granularity are discussed with their benefits and limitations. Based on this literature review, fuzzy-ontology based fine-grained AR approach is presented by extending the expressivity of the ontology model to gradually define imprecise sensor measurements and defining conditions for individual actions with a given object. Next, the system implementation details and evaluation results of the proposed fine-grained AR approach are presented. This chapter concludes with the discussion on the findings and future work on optimising fuzzy-based fine-grained AR approach.

Chapter 5 investigates on handling uncertainty factors affecting AR process to improve the accuracy and estimating confidence level when detecting ADLs. In this chapter, the nature of uncertainties and related work are initially analysed. Next, a probabilistic ontology-based approach is proposed to incorporate the uncertainty factors in the AR approach. Subsequently, the implementation and evaluation results are presented to illustrate the applicability of the approach in the AAL system. This chapter concludes with the findings and lessons learned from the proposed probabilistic ontology approach with a discussion on future work.

Chapter 6 presents a framework for incorporating fuzzy/ambiguous and uncertainty reasoning in a knowledge-driven single-user AR system. Chapter 6 initially define the concepts of fuzzy and uncertainty and the motivation to integrate fuzzy and probabilistic ontological model. Next, related work on handling uncertainty and vague observations from SH when performing AR, existing SH technologies and data storage technologies are studied. Based on the findings from the current studies, a semantic-enabled AAL system framework is proposed to model, analyse and store the data using semantic and SH technologies. The application of this framework is illustrated as a case study. This chapter highlights the key contributions made and provides a discussion on future work.

Chapter 7 focuses on developing multi-user AR approach at multi granularity level within a shared smart environment. Existing studies related to multi-user AR are analysed, and challenges in identifying users conducting ADLs and personalising assistance is highlighted. A multi-user AR approach is proposed within a shared smart environment is presented. The approach differentiates and describes the process of identifying ADLs conducted by single and multiple users by means of relationship between object and ADL description, time-series analysis and discriminative sensing attributes. Next, a multi-user AR algorithm and use case study have illustrated the applicability of the approach. Finally, a discussion and summary of the proposed multi-user AR are highlighted.

Chapter 8 analyse the challenges and opportunities in selecting suitable AAL system architecture based on current studies and state-of-the-art SH technologies. Next, two popular types of service-oriented architecture (SOA) for AAL system are described: multi-layered SOA and microservices system architecture (MSA). The evaluation and discussion of two types of SOA implementation are provided. This chapter concludes by highlighting key objects satisfied in this chapter and future direction to optimise the system to be suitable for real-world application.

Chapter 9 reflects upon the overall contributions made in this thesis. This chapter provides a summary of the research activities undertaken during this research project and sheds light on future research directions in developing AAL system to allow widespread adoption of the system for the private homeowners, commercial businesses and other stakeholders.

CHAPTER 2. HUMAN ACTIVITY RECOGNITION IN INTELLIGENT ENVIRONMENT

This chapter reviews the state-of-the-art and well-established studies carried out by the research community in relations to the building an AAL system. A typically AAL system comprises of several key components; they are activity modelling, data collection and monitoring, data processing, activity inferencing and recognition, aiding when required, dynamically learning and evolving user models, application type and human-computer interface (HCI). These components complement each other within the AAL system and have their strengths and limitations to enable a coherent solution to form. Figure 2.1 illustrates these components as a building block of an AAL system in a pyramid form. The diagram is read bottom-up whereby the core components such as activity modelling, and data collection lay the foundation for AR and the higher layers. Nevertheless, AR plays an important role in the AAL system to provide context-aware assistance and this chapter reviews these key components. In addition, state-of-the-art system architecture styles and patterns adapted for AAL systems are investigated in order to identify open challenges and issues for developing and deploying a real-time assistive system. Finally, a summary of this chapter and open issues are highlighted.

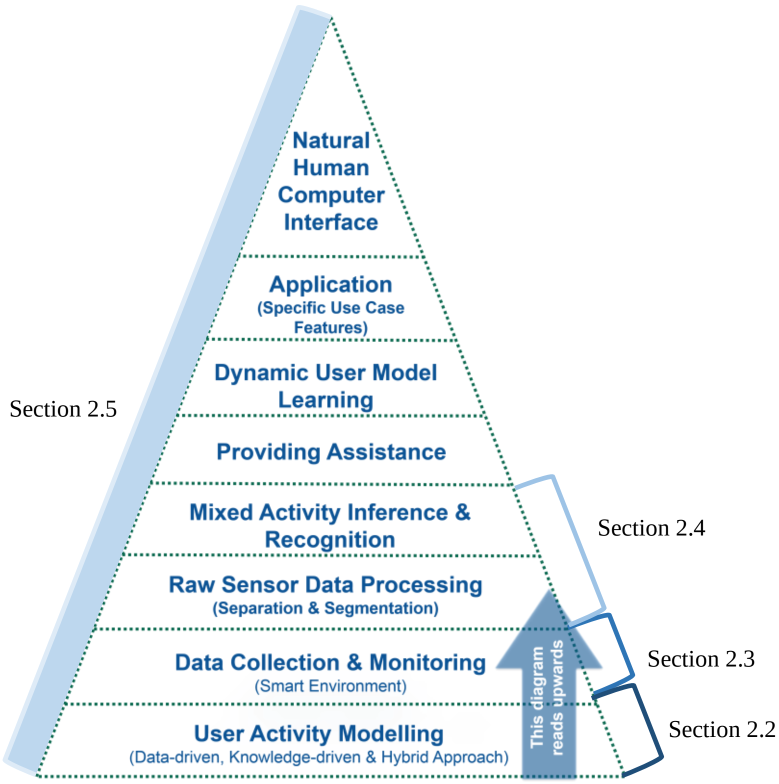


Figure 2.1. AAL System Building Block

2.1. Introduction to Human Activity Recognition (HAR)

Human activity recognition (HAR) can be defined as a process whereby an actor's behaviour and their environment are monitored and analysed to deduce the unfolding activity. The objective of the HAR is to DISENTANGLE the activities dynamically and learn their change in behaviour over time for accurate assistance. The idea of DISENTANGLE is motivated by the view that any activity is essentially the emergent behaviour of its constituent actions. For example, three activities can emerge from the following sequence of actions: *taking a kettle, turning the water tap, filling water, boiling water, taking a mug, taking a pan, taking teabag, turning on the cooker*, *adding hot water to the mug*, *adding hot water to the pan*, *adding milk, adding pasta*. This corresponds to activity "*preparing hot water*" and sequentially followed by the interleaved activities "*preparing tea*" and "*preparing pasta*".

2.1.1. Defining Single Activity, Mixed Activities, and Action Levels

Before dwelling deeper into describing the processes of HAR, it is important to recognise the nature of how one or more activities are performed in a real-world environment. Figure 2.2 depicts a general hierarchal structure of how one or more human activities can be performed by a single person (*A*) or with multiple people (*B*). A single person can perform one activity at a time interval sequentially or more than two activities at a time interval with their actions occurring interweavingly or concurrently. The actions for each activity are generally performed in any order and can be assumed independent of previous/future actions. However, some dependencies between previous and current actions can exist. A single-user can also work in collaboration with other users or independently in a shared space to complete one or more activities [28]. In collaborative HAR context, the data associated with a specific user to provide personalised assistance are some of the key challenges faced in shared users space [29].

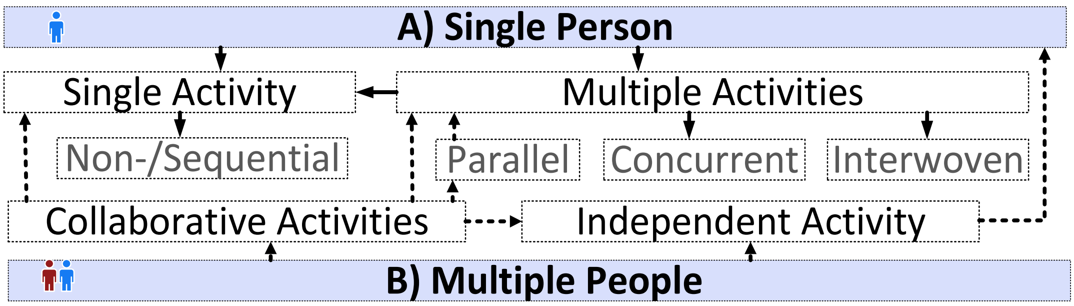


Figure 2.2. Types of activities performed by a single user (A) and multiple users (B)

Each action within a given activity can be analysed at multiple granularities based on available information; coarse- or fine-grained level. The coarse-grained level action recognition (AR) involves inferring relationships between an object being used, the object's relation with ADLs, user's location and time interval to assume a given activity is occurring. Whereas, fine-grained level AR analyses how the user interacts with the object and verifying with multimodal sensor data to determine the completion of intended action with the object of interest. This fine-grained level AR becomes particularly crucial with users suffering from physical disabilities, tremors or even forgetting to complete the actions due to decline in memory. These are some of the common symptoms of Alzheimer's and Parkinson's diseases commonly reported within the ageing population that affect their ability to conduct ADLs [30], [31].

2.1.2. Five Key Phases of HAR

The process of HAR can be described in five stages described in Figure 2.3. Firstly, the data collection stage is to take advantage of rapidly developing Internet-of-Things and ubiquitous sensing technologies to not only monitor environmental changes but also sense user's actions at the fine-grained level. There are heterogeneous sensing devices and platforms available and can be categorised as vision and sensor-based approaches. Whilst the vision-based sensing approach has been successfully applied in areas such as security surveillance, the sensor-based approach has become more attractive in the smart home (SH) environments due to lower ethical and privacy concerns. The sensor-based sensing approach enables a varied level of data collection methods; ambient, object embedded (or dense) sensing and wearable sensing [32]. The wearable sensing technologies can be further categorised as outerwear and implantable [33]. More details of data collection and monitoring approaches, challenges, and opportunities issues can be found in section 2.3. Nevertheless, due to such diversity in sensors and the type of contextual data being generated at different frequencies simultaneously, one inherent challenge is to segment the sensor data stream concerning the ongoing/new set of activities queue and support AR.

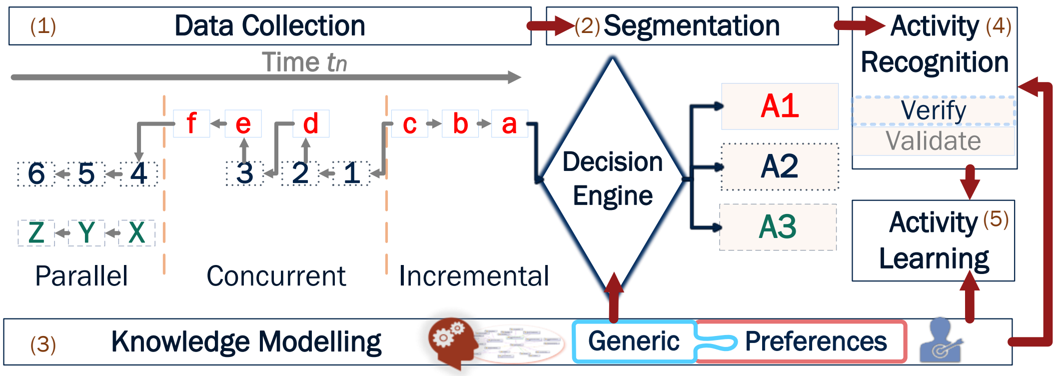


Figure 2.3. Five interdependent phases of AR: (1) data collection, (2) segmentation of sensor observations, (3) knowledge modelling, (4) AR, and (5) activity learning.

The second segmentation stage is therefore responsible for organising the observed sensor events based on the ongoing activities or detecting new activities performed by a single user in mixed activities scenario is a major challenge being investigated in this chapter. In order to make segmentation decisions, prior knowledge model is essential to verify association links such as what everyday object is the sensor attached to, contextual information (i.e., location, time and ambient attributes) of the object and relevance of object to which ADL(s). The set of segmented sensor observations for a given activity is later passed to AR algorithms to analyse the data to determine completion of intended actions and provide effective assistance when necessary. Hence, a correctly segmented set of sensors can boost the accuracy of the AR algorithm, performance and optimise the usage of computational resources.

The third stage is to develop a computational model to hold information such as ADLs, smart environment, and user preferences. The model is largely developed using DD, KD and hybrid approach. In the DD approach, large pre-recorded datasets are processed using generative or discriminative classification techniques to produce the activity model [18], [20]. In contrast, the KD approach relies on domain experts in the field of interest to conceptualise domain heuristics using various knowledge modelling tools. KD approach uses formal and logical theories to create a well-defined knowledge which is human and machine-processable, i.e., ontological models. Therefore, enabling KD approach overcomes the “cold start” issue by not processing a pre-recorded dataset, however, falls short in handling unseen or uncertain data [5]. Moreover, the common problem of DD and KD is that it assumes a complete description of all the ADLs, entities and concepts within the activity model. Consequently, the hybrid approach [34], [35] is used to combine the expressivity power from KD and the ability to learn patterns/frequencies, handle unseen or uncertainty in events from the DD approach to growing the initial model incrementally.

The last two stages, AR and activity learning approaches [34], are influenced by the selection of modelling approach and the quality of the segmented sensor data for reasoning. AR is described as a two-fold process: verification and validation. The verification process inspects the relationships between ADLs and a set of sensor observations, while validation process calculates a degree of confidence of actions occurring in a given activity. The role of the activity learning is to evolve the initial knowledge model by analysing the AR results and sensor observations to discover new activities, patterns, and user preferences in real-time or offline. The activity classification and activity learning topics are beyond the scope of this chapter; nevertheless, for more details, see [14], [19]. This chapter will mainly focus on verification phases of the activity classification process to reduce the computational complexity and time delay to incrementally grow the set of segmented data for a given activity as the events unfold.

Several human factors further increase the complexity when designing the knowledge model, developing segmentation and AR algorithms. As discussed in section 2.1.1, it is common that one can perform single or mixed ADLs at a given time, and this is illustrated in Figure 2.3. Individual ADLs (*A1*, *A2* and *A3*) can have a set of actions ({*abcdef*}, {*123456*} and

{*XYZ*}) which can be performed in any order. A single ADL (*A1*) can also be conducted in conjunction with multiple other ADLs; either incrementally (i.e. *A1* then *A2*), concurrently (i.e. *A1* with *A2*), and in parallel (*A2* and *A3* running simultaneously). Furthermore, people follow a specific tradition, ritual, culture or even have their own preferences to conduct basic ADL tasks which makes it difficult to generalised ADL description.

2.2. Activity Modelling

The activity modelling techniques have been generally classified as DD, KD and hybrid approach[19]. DD approach performs computation on the pre-collected datasets using various patterns, probabilities and statistical methods to identify and generate the activity model. In contrast, the KD utilises rich domain knowledge of human beings to create the conceptual activity models of the real-world using knowledge engineering techniques. These rich models are commonly created by a domain expert, and it can be easily reused and shared in comparison to DD approach, where a lengthy computation and data mining algorithms are required. Both of these approaches have their own strengths and weaknesses; further discussed in sections 2.2.1 and 2.2.2. The hybrid approach was later introduced, which utilises the strengths of DD and KD approaches to tackle some of its shortfalls; more details in section 2.2.3. Figure 2.4 provides a simple diagram to illustrate the three classifications. In the following sections, the aforementioned approaches are reviewed in terms of their nature, benefits and limitations, and the relative studies[36].

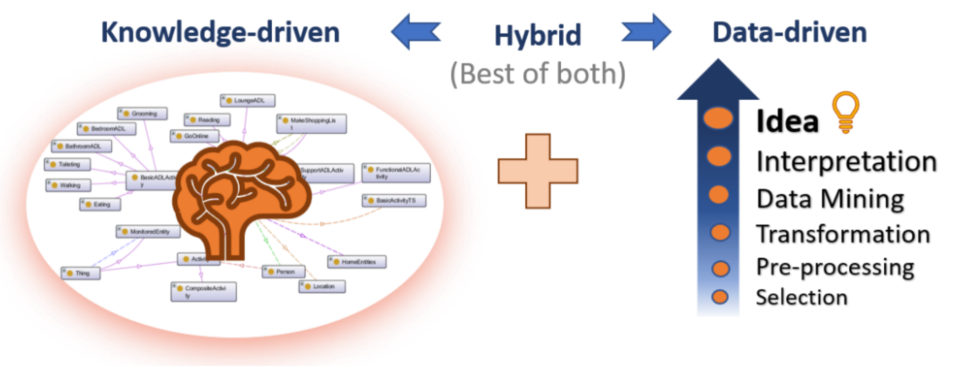


Figure 2.4. Activity modelling approaches: KD, DD and Hybrid

2.2.1. Data-driven (DD) Approach

DD based AR approach consists of analysing and training activity models from large pre-existing datasets. The key advantages of the DD approach are that it is highly sensitive to unseen data[37], supports modelling uncertainties and temporal knowledge. The techniques used for activity modelling in the DD approach are classified as a generative and discriminative approach[19]. The popular generative modelling methods are Bayesian networks, partial Markov decision process (POMDP)[38], a variation of Markov model to model action sequences as finite states with their transitional probabilities and continuous state-space model (CSSM) [39]. Whereas, conditional random field (CRF) and support vector machine (SVM) are widely used as discriminative methods to improve the accuracy and performance of the activity recognition[15]. On the whole, the generative approach attempts to build a comprehensive description of the input or data space, normally using probabilistic models such as Bayesian network. On the other hand, the discriminative approach only models the mapping from input data to activity labels as an output. The strengths and limitations of common methods adapted by generative, discriminative and combination of two approaches have been outlined in the table below. In addition, a description of some of the methods and critical related studies have been analysed in the three sections, 2.2.1.1, 2.2.1.2 and 2.2.1.3, respectively.

Table 2.1. Outline of the data-driven approaches and their classifications

	Generative	**Discriminative**	**Heuristic/Other approaches**
Description	Focuses on representing all the activities from the given input.	Further classify the generic activities and label data to the associated activity.	Combinations of both methods which makes it difficult to classify.
Common approaches	Naïve Bayes (NB), hidden Markov model (HMM) and dynamic Bayesian networks (DBNs) [40].	K-nearest neighbor(KNN), artificial neural network (ANN), support vector machine (SVM), distance learning (DL) and conditional random fields (CRFs) [40].	Heuristic rules and statistical models of sequential patterns, HMM model and the reactive planning engine, patterns by using the frequency of the sensor, diverse classification methods to analyse multimodal sensor data [40].
General evaluation	*Advantages* - Models are flexible as they learn structure and relationships between the classes by using prior knowledge for a given task (i.e. HMM) - Prior distributions and probability reasoning. - Performs well with uncertainty in data. *Disadvantages* - Parameters are not optimised - Require a large amount of data	*Advantages* - Models are computationally efficient. - Capture fine details - Remain robust in the prediction of class labels - The capability of tuning the parameters for the task at hand *Disadvantages* - Suffer from over-fitting - Require a reasonable amount of data. - Can contain limited diversity of training models.	*Advantages* - Flexible integration of classification methods suitable for domain problems. - Increase the accuracy of AR with classifiers analysing multiple features/ parameters. *Disadvantages* - Synchronising multiple methods. - Efficiency can be reduced with incompatible methods. - Incomplete activities data can affect the accuracy of AR results.
Overall	Suffers from expensive computation to be performed on the pre-collected datasets which create well known cold start problems and lack of completeness of the activity models.		

2.2.1.1. Generative Modelling

As discussed above, the generative modelling simply captures all the observations from the environment and creates a holistic model of the activities. Some of the popular approaches are naive Bayes[36], hidden Markov models (HMMs) [41], [42], Dynamic Bayesian Networks (DBNs), artificial neural network (ANN), and distance learning (DL)[40]. Some of these classifiers are briefly described below.

Likelihood *Class Prior Probability*

Posterior Probability $\quad P(c \mid x) = \dfrac{P(x \mid c)\, P(c)}{P(x)}$

Predictor Prior Probability

Figure 2.5. Bayes theorem: finding a posterior probability of the class (c) given predictors (x = data)

Table 2.2. Calculating the probability for the defect occurring at three TV manufacturing factories using Bayes theorem

A) Frequency Table

Factory	% of total production	Probability of defective TV (D)
A	0.35 = *P(A)*	0.015 = *P(D | A)*
B	0.35 = *P(B)*	0.010 = *P(D | B)*
C	0.30 = *P(C)*	0.020 = *P(D | C)*

B) Calculating Predictor Prior Probability

P(x) = P[(D∩A) U (D∩B) U (D∩C)]
= P[(D|A). P(A) + (D|B).P(B) + (D|C).P(C)]
= P[(0.015).0.35 + (0.010).0.35+ (0.020).P(0.30)]
= P[(0.00525 + 0.0035 + 0.006)]
= P(0.01475)

C) Calculating Posterior Probability

⇨ Probability of finding a defective TV (D) from **factory A**.

$$P(A|D) = \frac{P(D|A)\,.\,P(A)}{P(x)} = \frac{0.015\,.\,0.35}{0.01475} = \frac{0.00525}{0.01475} = \mathbf{0.356}$$

⇨ Probability of finding a defective TV (D) from **factory B**.

$$P(B|D) = \frac{P(D|B)\,.\,P(B)}{P(x)} = \frac{0.010\,.\,0.35}{0.01475} = \frac{0.0035}{0.01475} = \mathbf{0.237}$$

⇨ Probability of finding a defective TV (D) from **factory C**.

$$P(C|D) = \frac{P(D|C)\,.\,P(C)}{P(x)} = \frac{0.020.\,0.30}{0.01475} = \frac{0.006}{0.01475} = \mathbf{0.407}$$

The simplest generative approach used by the researchers is naïve Bayes classifier (NBC)[36]. The NBC is based on Bayes' theorem, which assumes predictors (data) being independent. The Bayes' theorem provides a method of calculating the posterior probability from the likelihood of an event occurring at a given location/class. Figure 2.5 provides an equation of the Bayes theorem with colour coordinated of the relevant section of the equitation. The equation can read as the probability of the class (c) given predictors (x = data) equals the probability of the predictors multiple by the probability of the class divided by the sum of the predictor prior probably. In general, this formula attempts to find a likelihood of an event occurring at a specific class from the probability of the whole datasets. Table 2.2 provides an example of Bayes theorem being applied to find the probabilities of the defect occurring in the TV from the three different factories of one manufacturing firms[43].

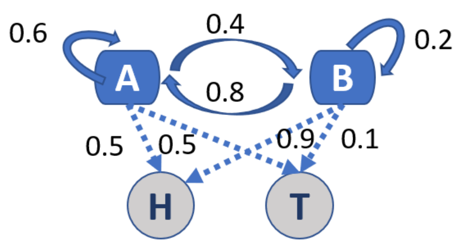
The NBC models can be applied to find the probability of a particular event occurring from the sensor observation. The NBC models take into consideration of all the sensor readings as an observation, and the activities are given discrete labels. These labels are based on the set of prior observations and the probabilistic function, which is used to estimate the likelihood of the activity. Although these classifiers assume conditional independences activities, it provides good accuracy even with the vast amounts of data. Furthermore, the simple nature of the classifier can outperform more sophisticated classification methods. One of the limitations of NCBs is that it does not explicitly support temporal information, which unfortunately is an important factor in activity recognition.

The work in [44], adapted naïve Bayes classifier on the ontology-based question and answering systems. The proposed algorithm was able to construct ontologies autonomously from the unstructured large-scale text using syntax and semantic probabilities. The algorithm iteratively extracted a list of attributes and relations for the given seed concept (manually parsing pattern rules) and a binary decision tree-based rule engine. The bespoke ontologies are created and updated on upon questioning and answering process. This approach was evaluated with benchmark datasets and the performance with gold standard and other well-performing methods. The result of the experiment indicated that the proposed method achieved higher accuracy in constructing a generic domain ontology.

Another popular classifier is the Hidden Markov Model (HMM), which can handle the temporal information. HMM is an information modelling tool that allows sequences of observations to be represented into a probability distribution model[12]. HMM is created a specific structure that enables the data to be quickly learned and interpret with the learned model; both are considered to be easy and efficient to implement. The HMM has two defining properties [12]. Firstly, it is assumed that the observation t was generated by some process whose state S_t is hidden from the observer. The set of hidden (latent) states in the classifier are coupled in stochastic Markov chain, in a way that states at the given time depend only on the values of states at the finite number of preceding times. The observation is then generated probabilistically through the stochastic (random) process. Secondly, it is assumed that the hidden state of the previous observation (S_{t-1}) and current observation (S_t) are independent (Markov property). This classifier played a major role in speech recognition literature, hidden states were linked to phoneme labels and the features extracted from the audio data are recorded as observations. HMMs based modelling has been recently adopted by computer vision for modelling sequential data (video). Furthermore, the HMM approach uses MarkOv chain instead of discrete set of states.

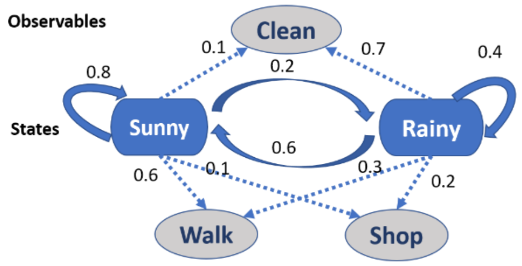
Table 2.3 provides two simple examples HMM being applied in a real-life scenario; fair/biased coins (a) [45] and weather-based Markov chain (b).

Table 2.3. Two HMM examples a) fair/unbiased coin and b) weather.



Hidden States ➔ Coin types: F= fair coin, B = biased coin.
Observations ➔ Sides : H = head, T= tail.

a) Fair and Biased Coins



Hidden States ➔ Weather: Sunny and Rainy.
Observations ➔ Activity: Walk, Shop and Clean.

b) Weather-based Markov Chain

The HMM approach has been adopted by other models such as a linear dynamical system (LDS), commonly known as the Kalman filter, which uses continuous states. LDS has been used for physiological condition monitoring systems with a variety of sensors data used as an input which was also introduced to handle unmodeled variations in data; being one of the major shortfalls of the generative approach.

HMMs also enable one to create a statistical temporal model. They are a special case of general Dynamic Bayesian Networks (DBNs), which are Bayesian networks in which discrete-time index is explicitly represented. Inference and learning in DBNs are performed simply by using network propagation in Bayesian networks; usually making a Markovian assumption with explicit representation of conditional independences in variables. The popular use of DBNs for activity monitoring was in the Lumiere project for modelling user's assistance needs to be based on their activities on the screen.

Coupled HMM (CHMMs) is an extension of simple DBNs for recognising simultaneous human actions. CHMMs have two Markovian chains for "*modelling different streams of data with a coupling between them to model their interferences*"[36].

Hierarchical Hidden Markov model (HHMM) is derived from HMM, which inherit not only the probabilistic nature but also the hierarchical based hidden state structure for activity modelling. The parent node is known as an abstract state and the last child node as a production state. The observations are appended to the last production states, and at individual levels, an end state is introduced to represent the completion of activation for the child node; for further details on HHMM, see [41],[42]. The work in [23], adapted the HHMM and joint probabilistic data association filters (JPDAF) approach for activity modelling and Rao-Blackwellised particle filters (RBPF) for approximate inferencing. This method enabled them to recognise complex activities being performed by multi-users.

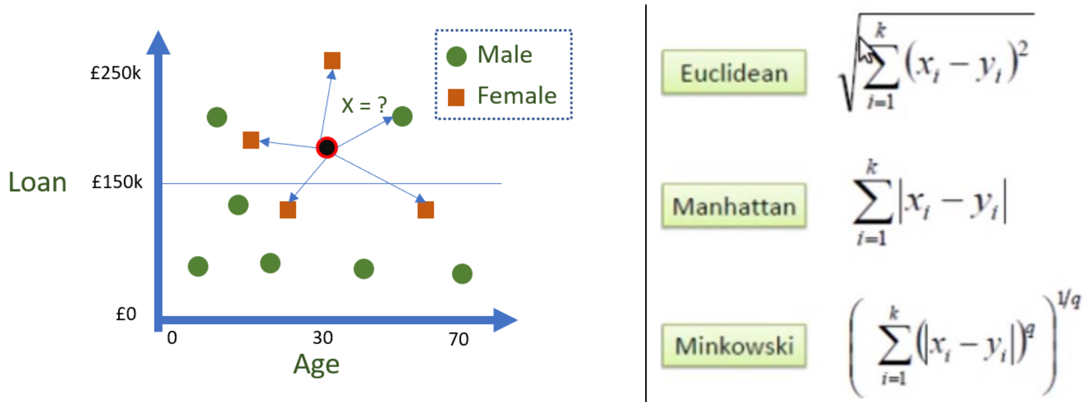
The work in [46], proposes a SACAAR system architecture based on context-driven activity theory (CDAT) for recognising complex activities. It adapts probabilistic and Markov chain analysis approach to discover mixed activity signature and generate definitions for the mixed activity. The activities are decomposed into atomic activity, and the contextual data of the data is analysed to infer any association with other complex activity. The complex activity recognition (CAR) algorithm proposed in the paper achieved the overall accuracy of 95.73%, reduced inferencing time to 32.5% and the training data required by 66%.

Despite the fact that HMMs and DBNs are simple and popular, they do have some limitations. An HMM is not capable of capturing long-range or transitive dependencies of the observations, mainly due to its very strict independence assumptions (on the observations). In addition, without adequate data size and training, HMM may not be able to recognise all the possible observation sequences for a given activity.

2.2.1.2. Discriminative Modelling

The discriminative approach focuses on the classification problems in comparison to a generative modelling approach, which concentrates on representing the complete description of the sensor observations [36]. The primary objective of the discriminative approach is to further analyse the sensor observations that were generally or implicitly described the generative approach. Some of the common discriminative approaches are rule-based (heuristics) approaches, neural networks, conditional random fields (CRFs), and linear or nonlinear discriminative learning (i.e. support vector machines (SVN)) [36].

The simplest and popular discriminative approach is the Nearest Neighbor (NN), also known as k-Nearest Neighbor (KNN). This approach uses novel sequences of data from the existing data set, and the K number of nearest points of the data is compared to get the majority of the vote to determine the activity labels[47]. More generally, it classifies the activity according to the majority of the vote from the K nearest points in the data set using distance functions, i.e. Euclidean (popular), Manhattan, and Minkowski (see Figure 2.6 (b)). The value for the K can be any prime number to avoid getting equal results. For instance, if K=1, then use the first closest nearest label from the graph, similarly if the K=5, then we take the majority vote from the five nearest labels, i.e. if we have two males and three females, then classify the value of x to be male. The common practice, however, is to select the K value between 3-10 has produced better results than 1NN. However, choosing the optimal K value depends on the data type and its size. In general, larger the K value, more accurate the result as it reduces the overall noise; nevertheless, there is no guarantee. Another approach is to use cross-validation method which uses the independent dataset to validate the K value.



a) Nearest value in the plot graph when K=5. b) Distance Functions

Figure 2.6. Nearest Neighbor (NN) plot graph (a) and distance functions (b)

In general, the KNN is a popular algorithm for pattern recognition and is par with decision tree in terms of performance and the computational complexity[47]. However, in the comprehensive study[48], KNN and decision tree algorithm (J48/C4.5) were evaluated using accelerometer data in different experimental settings, and the result indicated that KNN achieved higher accuracy overall. These results were further backed by the study in[49], where heart rate and accelerometer data were used to recognise different sets of activities.

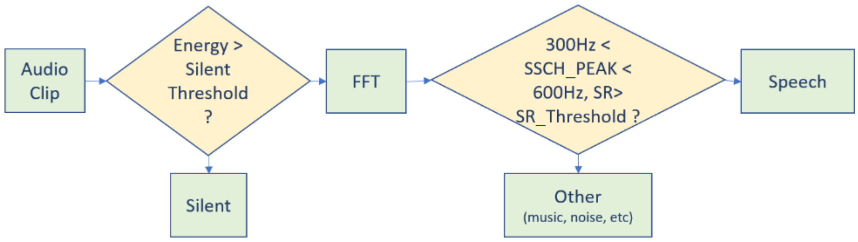
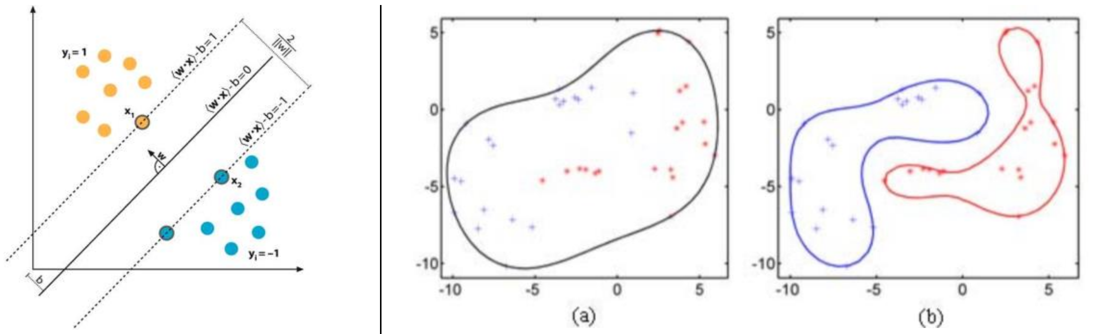


Figure 2.7. An example of a decision tree for background sound classification.

The decision tree is another approach which partitions the training data into subsets of the relevant activity and a set of rules. This approach allows rules to be generated that is understandable by the user; however, it is often difficult with large numeric data that require high-precision. Other key advantages of using a decision tree approach are that it offers low complexity in implementation and excellent interpretation. Hence, it has been adapted and used as a main classifier in many activity recognition studies. For example, work in[50], decision tree-based algorithm to classify sound, location and accelerometer information of user activities. Figure 2.7 illustrates how the algorithm was applied to the audio clips from the mobile microphone component to infer loud, silent and speech environment. One of the disadvantages of the decision tree is that once the model has been built, it is costly to update the model with the new training datasets. Hence, the decision tree classifiers are not popular with the online learning strategies for AR[47].

Weka Toolkit [51] is a Java-based program where many machine learning algorithms exist and has been used in many research studies[47]. One of the decision tree algorithm C4.5 is named as J48 in Weka. J48 algorithm has been used in several AR research studies as an offline classification model.

Several discriminative approaches explicitly investigate the data points closest to the boundary that is of interest; known as "hard" data points. These "hard" points play a significant role when classifying different activities. In comparison to other "easy" data points which are more away from the boundary are considered as less relevance. Therefore, the challenge is to analyse these "hard" data points, known as "support vectors" in the support vector machine (SVM). An SVM is another machine learning technique to identify the support vector points automatically. This technique has been compared with five other classifiers in [52] with the four publicly available smart home datasets; these techniques are SVM, Evidence-Theoretic KNN (ET-KNN), Probabilistic Neural Network (PNN), KNN, and NB. The result indicated that SVM and ET-KNN outperformed other activity recognition methods. The support vector data descriptors (SVDD) classifier, a variance of SVM, was introduced to describe the target data set in a spherically shaped boundary[53]. This method was further optimised such as work in [54] where the hyper-spherically shaped boundary was introduced and the mixture of SVDD (mSVDD) [55], where a statistical method known as was Expectation-Maximization (EM) was introduced to train the model. In this work[56], the SVDD classifier was applied to detect the normal and anomalies behaviour patterns of the elderly. Figure 2.8 illustrates how SVM data points at the boundary are analysed and two variances, SVDD and improved SVDD.



i) SVM: Support vector data points and boundary	ii) (a) SVDD and (b) improved SVDD using hyper-spherically shaped

Figure 2.8. Illustrating the Support Vector Machine (SVM) classifier and some its variances

In a real-world, several activities can have non-deterministic nature, i.e. the sequences of the tasks, concurrent or interwoven. A conditional Random Field (CRF) approach was introduced to address this issue; an alternative option to HMM for higher flexibility in terms of state independence assumptions and acyclic nature of capturing relationships from the observations. The CRF approach can be classified as both a discriminative and generative probabilistic model that represents the dependence of hidden variable y and observed variable x[36]. Both HMMs and CRFs are used to predict the current activity based on current and previous observations. However the key difference is that HMM attempts to join the two probability distribution p(x,y), whereas CRF attempts to predict the current activity using conditional probability p(x|y), i.e. P(adult | age > 18). Also, CRF achieves its flexibility by allowing arbitrary and non-independent relationships among the observations sequence. Furthermore, the CRF approach relaxes the independence assumptions, where the hidden probability may depend on the past and even upcoming observations.

A CRF is modelled as an undirected acyclic graph (graph with no cycle or set direction, see Figure 2.9). It has a flexibility to capturing any relation between observation and hidden state.

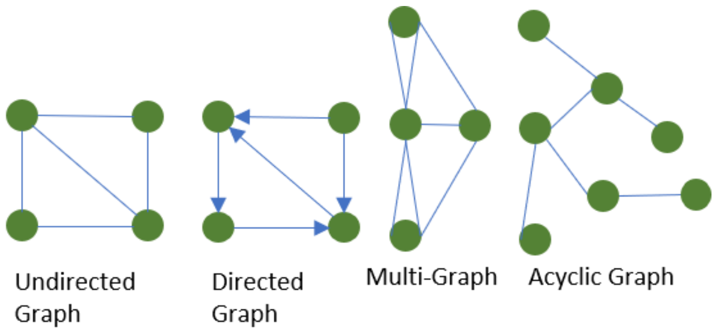


Figure 2.9. Illustrating the acyclic nature of the Conditional Random Field (CRF)

2.2.1.3. Heuristic/other approaches

Many approaches that do not fall clearly into discriminative or generative categories[36]; rather, the combination of these approaches are used along with some heuristic information. For instance, the work in [40] employee combines both generative and discriminative methods to perform activity recognition. The distance minimisation (DM) is used as a discriminative method which measures interclass feature distance by its mean representations for an individual class. For the generative method, the probability estimation (PE) method has been used to measure the actual distribution of the obtained distance by curve fitting technique. The benefit of using DM method is that it avoids decision biases towards the class, which has the majority of instances, but it can result in over-fitting. The term over-fitting is the term used when there is either limited training data available or imbalanced number of activity instances. Although the PE method has been employed to calculate the probability of the class make the generalised and unbiased, it would require larger training datasets. In addition to the DM and PE methods, the oversampling technique is applied when required and SVM classifier to map the output of DM and PE methods; further generalising the results. This approach outperformed the two baseline approaches DM, PE; two learning approaches evidence-theoretic K-nearest neighbor (ET-KNN) and pairwise nearest neighbour (PNN), and several other state-of-the-art recognition approaches.

2.2.2. Knowledge-driven (KD) Approach

KD activity modelling is inspired from real-world conceptualisation and observations of activities which may or may not require objects interactions; the number of objects required for a single activity is limited and functionally similar despite performing the activities in varying sequences. For instance, a simple and very common activity such as "*making coffee*" which require following objects to make it, coffee pot, hot water, a cup, coffee, sugar and milk; "*brushing teeth*" require toothbrush, toothpaste, water tap, cup and towel. However, due to the nature of human beings, everyone has a different lifestyle, culture, habits, preference, or ability to perform various activities in different ways, i.e. one may prefer strong white coffee, other with a specific brand of coffee or different types of milk (skimmed or whole) and make in a different order, i.e. adding milk first then water or vice versa. Hence, this kind of domain-dependent and activity-specific prior knowledge provides valuable understandings into how activities can be constructed in general and how they can be performed by individuals in a specific situation.

However, human beings have a different lifestyle, culture, habits, preference, or ability to perform various activities in different ways, i.e. one may prefer strong white coffee, other with a specific brand of coffee or different types of milk (skimmed or whole) and make in dissimilar order, i.e. adding milk first then water or vice versa. Furthermore, human beings have some optional activities which they want to perform as an alternative or in combination, i.e. using mouth wash and/or using toothbrush and toothpaste for "*brushing teeth*" activity and different ways of heating water, i.e. using a kettle or on the hob and a stockpot (possible preparing for multiple people).

The rational of KD modelling is to make use of knowledge engineering methodologies and techniques, to acquire domain knowledge and encode it in a various reusable knowledge structure. This includes activity modelling (containing heuristic and prior knowledge), context models (which has relationships between activities, objects and temporal) and spatial contexts defined. The domain knowledge is captured, represented for activity modelling and recognition can generally be classified in mining-based (using existing knowledge publicly available), logic-based (rule-based) and ontology-based (philosophical way of formally representing real-world axioms). Also, the knowledge structures can be modelled and represented in different forms, i.e. schemas, rules or networks.

The semantic web technologies are the backbone of the knowledge-driven approach due to its capabilities of formal knowledge representation, storing, querying, manipulating, reasoning, exchanging and programming with raw sensor data. The recent survey in [57], presented a comprehensive overview of semantic web technologies and the current open issues and challenges of using this technology. In general, it provides an overview of formal knowledge representation methods, modelling concepts such as sensor data, context and events, reasoning frameworks, inferring and reasoning from the sensor data stream, handling event uncertainty and the challenges of this approach.

2.2.2.1. Mining-based approach

The mining-based approach adopts existing available data source to avoid "cold start" problems in comparison to DD approach where it suffers from both, the "cold start" and reusability of the models[36]. Although these existing data sources are still required to be analysed and processed to create a probabilistic or statistical model. The general process of mining-based approach can be in the following sequence:

1. Identify activities and relevant sources (objects)
2. Information retrieval and analysis techniques
3. Algorithms, probabilistic & statistical analysis methods such as occurrence and association to estimate object-usage probabilities.
4. Creating an activity model such as HMM using mined object and usage information for activity recognition.

In general, the approach attempts to extract object's usage information to deduce their related usage via a probabilistic method, i.e. "*mug*" and "*teabag*" object to a given activity named "make tea" [36]. In the past, Intel Research group initiated investigating on web mining, where they introduced QTag tagger system which analysed different websites to create a total of 21 300 activity models based on DBN approach[17]. The work in[58], employed adapted discriminative method (Viterbi algorithm and maximum likelihood) on the generic model and the Kullback-Leibler divergence technique to find similarities in the activities. In addition, work in [59], addressed similar object terms in different models using WordNet for synonymous words and segmentation problem for a sequential activity using the frequency of objects and discriminatory key objects in different activities [60].

2.2.2.2. Logic-based approach

The rationale of logical-based approach is to exploit the formal representation of the logical knowledge of a given activity, sensor data modelling and to use logical reasoning to perform activity recognition. The researchers have integrated various activity theories to create logical rules based on the models such as situation theory, lattice theory (Description logic (DL)), and event theory (event calculus (EC) [61]). In general, procedure creating a logical-based model includes:

1. Logical formalism to explicitly define and describe a library of activity models for all possible activities in a domain
2. Aggregate and transform sensor data into logical terms and formula
3. Performing logical reasoning, e.g., deduction, abduction and subsumption

High-quality ontologies are important for many applications. The Description Logics (DLs) have been recognised to be an ideal candidate as an ontology language in the past. However, there were restricted expressivity features and limited collection of knowledge-based ontology models. Nevertheless, recent research in DLs have aided to bridge this gap. The suitability of DLs as ontology languages has been highlighted by the inclusion in the several web ontology languages (OWLs), including OWL (OWL DL in specific). OWL is based on the resource description framework (RDF) schema syntax, which uses DL SHIQ to achieve a balance between expressiveness and the complexity level for the reasoning. Although the SHIQ presents high complexity for decision-making problems, many reasoning engines such as FaCT++, RACER, and Pallet have been used with impressive results. Therefore, allowing one to describe countless numbers of real-world facts as a set of rules. The inference engines can then be applied to deduce implicit knowledge data from the explicitly represented knowledge data model. The inferencing engines apply the set of rules to all the relationships, classes, methods and objects and instances.

Description Logics (DLs) are a family of knowledge representation language to explicitly represent the concepts and relations in a structured and formal means. For example, let's assume the following relationship "*A man that is married to a female who is an accountant has at least three children together, and all of whom are a musician*". Figure 2.10 describes the above notion in a simple statement. It uses various formal notations such as the conjunction (∩), negation (¬), the existential restriction constructor (∃R.C), the value restriction constructor (∀R.C), and the number restriction constructor (≥ n R). To apply this rule, let's say Bob is married to Alice, who is an accountant, and all of their three children are musicians.

$$\text{Human} \cap \neg\text{Female} \cap \exists\text{married.Accountant} \cap (\geq 3\ \text{hasChild}) \cap \forall\text{hasChild.Musician}$$

Figure 2.10. An example of Description Logics (DLs) statement using formal notations

Similarly, more axioms can be defined using Description Logic Program (DLP) and *ALCIO* TBox syntax; see Table 2.4 for more details.

Table 2.4. More example of representing axioms using DLP and ALCIO TBox syntax

Simple axioms	Description Logic Program (DLP)	*ALCIO* TBox
(1) Every man or woman is an adult	$\mathrm{Man} \subseteq \mathrm{Adult}$ (1) $\mathrm{Woman} \subseteq \mathrm{Adult}$ (1)	$\mathrm{Man} \cup \mathrm{Woman} \subseteq \mathrm{Adult}$ (1)
(2) A grown-up is a human who is an adult	$\mathrm{GrownUp} \subseteq \mathrm{Human}$ (2) $\mathrm{GrownUp} \subseteq \mathrm{Adult}$ (2)	$\mathrm{GrownUp} \subseteq \mathrm{Human}\ \ \mathrm{Adult}$ (2)
(3) A woman who has somebody as a child is a mother	$\mathrm{Woman} \cap \exists \mathrm{childOf}-.\mathrm{T} \subseteq \mathrm{Mother}$ (3)	$\mathrm{Woman}\ \exists\ \mathrm{childOf}-. \subseteq \mathrm{Mother}$ (3)
(4) An orphan is the child of humans who are dead	$\mathrm{Orphan} \subseteq \forall \mathrm{childOf.Dead}$ (4) $\mathrm{Orphan} \subseteq \forall \mathrm{childOf.Human}$ (4)	$\mathrm{Orphan} \subseteq \forall \mathrm{childOf.(Dead} \cap \mathrm{Human)}$ (4)
(5) A lonely child has no siblings	$\mathrm{LonelyChild} \subseteq \forall \mathrm{siblingOf}.\perp$ (5)	$\mathrm{LonelyChild} \subseteq \neg\exists \mathrm{siblingOf.T}$ (5)
(6) AI researchers are employed by the IBM	$\mathrm{AIResearcher} \subseteq \exists \mathrm{employedBy}.\{\mathrm{IBM}\}$ (6)	$\mathrm{AIResearcher} \subseteq \exists \mathrm{employedBy}.\{\mathrm{IBM}\}$ (6)

In general, the reasoning can be performed in many ways; the two common approaches are consequence-based and tableau-based. The consequence-based approach which uses horn fragment, whereas the tableau-based approach computes the classification from the given completion rules to infer additional facts. The logic ALC is one of the basic logic, which can be further extended for better expressivity, i.e. SHOIN. More detailed information about the family of DLs can be obtained from [62]–[65].

Furthermore, a knowledge representation system based on DL has two main components, terminological axioms (Tbox) and assertions formalism (Abox). The Tbox is a terminology used to define concepts and roles definition (i.e. classes, properties, and relationships in the ontology), whereas Abox can be used to describe an individual or the class by enumerating the individual instances[62], [66]. The Tbox can be used to introduce names or an abbreviation for a complex description. For example, we could introduce the abbreviation HappyMan for the concept described above. More expressive terminological formalisms allow the statement of constraints such as $\exists \mathrm{hasChild.Human} \subseteq \mathrm{Human}$, which says that only humans can have human children. Abox can be used to state the properties of individuals. For example, the assertions $\mathrm{HappyMan(BOB)}, \mathrm{hasChild(TOM, JESS)}$ state that Tom belongs to the concept HappyMan and that Jess is one of his children. A set of such assertions is called an ABox, and the named individuals that occur in ABox assertions are called ABox individuals[62], (See Table 2.5).

The logical rules are now being combined with ontologies to formally represent knowledge. For instance, RuleML language is being used in [67] this study to represent points of interest to a targeted group. In addition, semantic web rule language (SWRL) was created with the combination descriptive logic and production of external logic, i.e. OWL DL, OWL Lite, and Rule ML, see more [65], [68], [69]. In general, the logical-based modelling and reasoning approach has its own benefits and limitations. This has been summarised in Table 2.6.

Table 2.5. Example of Terminological axioms (Tbox) and Assertions formalism (Abox)

Tbox axioms	Abox formalism
∃hasChild.Human ⊆ Human	*HappyMan(BOB), hasChild(TOM, JESS)*

Table 2.6. Key strengths and weakness of knowledge-driven logical-based modelling and reasoning

Logical-based modelling and reasoning	
Strength	*Weaknesses*
- Semantically clear and elegant for reasoning. - Easy to incorporate domain knowledge and heuristics for activity models.	- Ability to represent fuzziness and uncertainty. - Minimal support for measuring the efficiency of the models - Lack of learning ability and the evolution of the rules.

2.2.2.3. Ontology-based approach

The ontology-based system relies heavily on formal representation and conceptualisation of the real-world axioms using semantic web technology. The term ontology can be described as an explicit, unbiased and unambiguous specification of a human knowledge conceptualisation [62], [70]. Nevertheless, human knowledge is an interaction of real-world truths and beliefs of wider communities, as depicted in Figure 2.11. Therefore, ontology modelling enables human knowledge to be formally represented, interpretable, processable, shared and re-used across multiple domains. An ontology consists of a set of concepts, relations (properties), instances and axioms (established or accepted statements) representation(C,R,I,A)[71]. In addition, due to the conceptualisation of a domain is formal, it allows the computer to perform inferencing and reasoning to derive additional information.

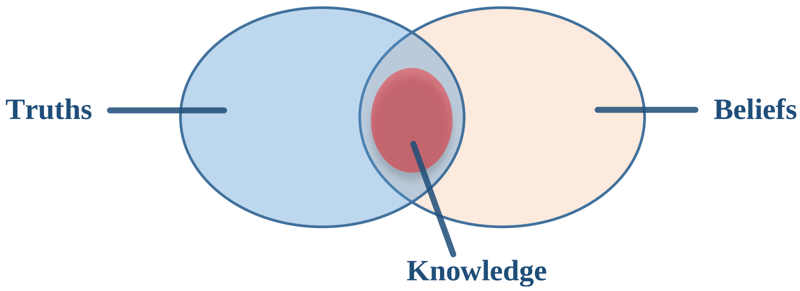


Figure 2.11. Definition of knowledge concerning truths and beliefs

An ontology model can be developed using the editing tool such as Protégé. The models can be created to the varying levels of abstraction, enabling one to encapsulate a particular set of knowledge. These expressive ontologies can then be shared and reused across different domains as a vocabulary. In this way, the model does not only get semantically enriched but also increases the reusability of the data, infer implicit information and perform reasoning using real-world axioms. Table 2.7 list some popular vocabularies that can be reused within any domain.

Table 2.7. Some popular vocabularies publicly available to reuse and share

Vocabularies	Formal upper-level Ontologies
- Friend-of-a-Friend (FOAF) - The Dublin Core (DC) - Socially Interconnected Online Communities (SIOC) - VCard (description of people and organisations) - RDF, RDF-Schema - Web Ontology Language (OWL) - Rule Interchange Format (RIF) – Logic rules to be exchanged between rule systems - RIF-BLD (the Basic Logic Dialect of the Rule Interchange Format) - Evaluation and Report Language (EARL) – for specifications, technical notes and testing results - Simple Knowledge Organization System (SKOS) - Good Relations (products sold online) - The Music Ontology - MarineTLO - Semantic Sensor Network (SSN)	- Basic Formal Ontology (BFO) - CIDOC Conceptual Reference Model - DOLCE - Descriptive Ontology for Linguistic and Cognitive Engineering - GFO - General Formal Ontology - OCHRE - Object-Centered High-level REference ontology - SUMO - Suggested Upper Merged Ontology - Business Objects Reference Ontology - YAMATO - Top ontology with objectives similar to those of DOLCE, BFO, or GFO

Table 2.8. A general overview of an Ontology-approach

Features	Ontologies
Vocabularies	Can be categorised in various forms, i.e. Domain, Upper and Hybrid ontologies. The most common are Web Ontology Language (OWL), RDF/-Schema, Friend-of-a-Friend (FOAF), The Dublin Core (DC), Simple Knowledge Organization System (SKOS) and so on. (NB: Many more can be viewed here: http://lov.okfn.org/dataset/lov)
Query language	SPARQL Protocol and RDF Query Language (SPARQL)
Logical rules	SPIN, SWRL, RuleML and others
Storage	Triplestore optimised to store RDF data, i.e. Jena Fuseki Server, NoSQL
Serialisation Formats	Triplets are represented in various formats such as turtle, N-triples, N-quads, JSON-LD, notation3 (N3) and RDF/XML.
Development tools	Protégé, triplestore(i.e. Jena Fuseki Server), reasoners (i.e. pellet, hermit and fact++)
Visualisation type	Graph-based

The work in [24] presents a hybrid method, system architecture, models, algorithms to recognise mixed activities and a multi-agent system prototype. The hybrid method combines the ontological and temporal knowledge representation formalisms capabilities. A generic conceptual activity model was developed to encode the characteristics of simple and mixed activities. The semantic web rule language (SWRL) was used to implement entailment rules for mixed activities. The designed rules further categorised in three perspectives: to derive interval relations and assert dynamic mixed activities, assert instances of fluent property, and derive/assert static mixed activities. An example of the overlapping activity and its relationship was presented in the SWRL rules. The activity recognition algorithm first segmented the data, and then the activity recognition module inferred simple and mixed activities with the help of SWRL rules and activity models. The system prototype was built using Java Agents

Development Framework (JADE) with Protégé for ontology editing tool, Java Expert System Shell (JESS) translator for SWRL rules inferencing, and Pellet reasoner. Finally, the evaluation result indicated that the system achieved an 88.26% for simple and mixed activities respectively.

2.2.3. Hybrid Approach

The hybrid approach is used to take advantages of different techniques introduced in DD and KD approaches. Many researchers have tried many combinations of approaches to improve or solve varying challenges, i.e. data modelling, processing and inferring activities.

The work in [72] proposes model namely, SC^2, which is a multi-layered activity modelling to represent four types of activities and employees time series shapelet-based approach to perform activity matching and recognition. This approach is capable of recognising simple, sequential, concurrent and mixed activities (sequential and overlapping) as the activities are decomposed at an atomic level (lowest level possible). The prototype was developed and tested using two open datasets for atomic activity and two case studies (simple ADL and basketball play). It was later evaluated using other three common approaches; Interleaved Hidden Markov Models (IHMM), Skip-Chain Conditional Random Fields (SCCRF) and Interval Temporal Bayesian Network (ITBN). Their results indicated a positive activity recognition for mixed activity. However, one of the limitations of this approach is that it requires shapelets to be trained, requiring a large amount of data to be processed.

2.2.3.1. Dynamic Activity Model Learning

One of the challenging aspects in the assistive system is to design a system that leverages the experiences from the previous tasks into a new task which has not been encountered before [19]. More informally, dynamically learn from the given observations that are outside of any explicit training data or activity models. Therefore, transfer learning term was introduced to represent the new task being drawn from a different population than the old [19]. This approach aims to provide many advantages such as reduced time spent processing large datasets, less information from human experts are required, and more situations can be handled effectively.

2.2.3.2. Machine Learning

The machine learning techniques have two types of activity learning methods, supervised and unsupervised. These methods mainly use probabilistic and statistical reasoning. The key difference between supervised and unsupervised learning methods is that supervised learning requires the data to be pre-labelled to learn and classify unknown data, whereas unsupervised methods process the unlabelled data[73]. However, other researchers have also classified the labelling approaches in other ways[19]. One of which is informed and uninformed, when the data is differentiated either by source, or target labelled data availability. Another is inductive (requiring labelled data), transductive (no labelled data required), and unsupervised learning methods. These perspectives have been reflected in Table 2.9.

Table 2.9. Data labelling approach from two perspectives: supervised vs unsupervised and informed vs uninformed

Data Labelling	**Supervised**	**Unsupervised**	**Common terms**
Informed	<u>*Informed Supervised (IS)*</u> Some data are available for **both** of the target and source domains.	<u>*Informed Unsupervised (IU)*</u> Data are **only** available in the target domain area.	<u>*Inductive learning*</u>
Uninformed	<u>*Uninformed supervised (US) **</u> Data are **only** available in the source domain.	<u>*Uninformed Unsupervised (UU) ***</u> **No** labelled data available in the target and source domain area.	<u>**Transductive /*</u> <u>***Unsupervised learning*</u>

The inductive attempts to learn the objective predictive function from the labelled data. The transductive view the relationship between instances, which does not always require labelled data.

The work in [19] provides a survey on transfer-based learning approaches. It categorises the transfer-based learning approach in four ways, sensors modality, by the differences in source and target environments, data availability, and the type of information being transferred. It further highlights researches carried by the types of knowledge being transferred concerning sensor modality and the data labelling process. From the grouping of the different studies in a table, it was clear that limited studies have been carried out in IU and UU data labelling/learning process and the relational knowledge transfer types. Furthermore, it sheds some light on the teacher/learner approach to transfer the knowledge in the realms of activity recognition. This model aims to presents a mechanism where the teacher teaches the learner on how to infer and learn the activity. The teacher can switch the roles by becoming a learner to learn new activities and obtain higher expertise. One limitation of this approach is that the learner depended on the teacher and therefore, the accuracy of the learner is also restricted by the expertise of the teacher. Also, the question remains unexplored in the situation whether the learner can exceed the teacher's capability and, in this case, how can the learner convince its superiors.

2.2.3.3. Genetic Algorithms

The work in [74] presents a reactive pull system (based in just-in-time philosophy) that use genetic programming and simulation tool to learn how to make decisions, i.e. generating a decision logic depending on the specific situations. The system aims to extract knowledge from the real-time observations, and from the system's current state as input, a decision tree can be created, and the suggestions in logical form can be returns as an output. This approach claims that no training set is used, and the knowledge can be represented as decision logics in a decision tree form autonomously. However, one of the limitations is that learning efficient decision strategies from the simulations tool can be computationally expensive. Therefore, this process is likely to be performed offline, and the resultant decision logic can then be used for online decision making. The benefits presented in this work can be further potentially exploited in the realms of activity learning and recognitions over a period of time. For instance, analysing the user behaviour over time in a separate offline environment and the out results be fed back into the online learning system.

2.3. Data Collection and Monitoring Approaches

Wired and Wireless Sensor Network (WSN) technology has enabled a large variety of applications to be developed; these have also been applied across many domains, i.e., military [75], healthcare, transport[76], and smart city infrastructure. WSNs play an important role in emerging Network-of-Things(NoT) or IoT paradigms [77]. The capabilities of the WSNs within the assistive systems can be seen as a supporting tool to allow humans or machines to interact with their environment and react to real-world events[78]. Therefore, the key responsibility of WSNs is to acquire environmental data from remote nodes and execute commands instructed by a coordinator, also known as a sink or base station. Depending on the application requirements, various communication protocols are available, through which a remote node can send data to the coordinator. These protocols have their own properties, benefits, and limitations, but they can be characterised by their range and energy consumptions. Some of the popular protocols are ZigBee [76], Z-Wave, WiFi, 6LoWPAn, 2G/3G/4G/5G, Bluetooth(BLE), radio frequency identification (RFID), near field communication (NFC), and infrared.

2.3.1. Sensing Approaches

The methods to collect data in a smart environment can be categorised into a vision and sensor-based sensing. However, in recent studies, researchers are using a multimodal approach, which combining these two approaches to retrieve fine-grained and meaningful data for higher accuracy in inferring user activity. Figure 2.12 depicts these approaches. The following subsections provide an overview of each aforementioned approaches and the related work that was carried out recently when performing AR.

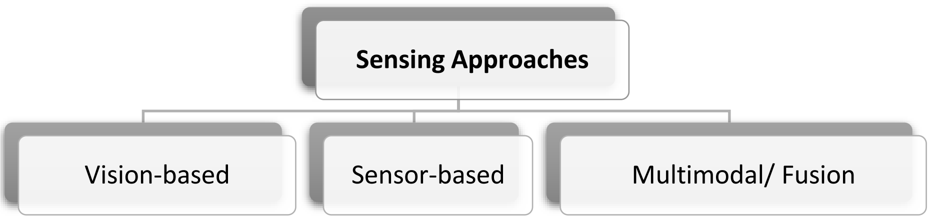


Figure 2.12. A categorisation of data collection/monitoring approaches

2.3.1.1. Vision-based approach

The vision-based sensing uses computer vision techniques to perform video-based and still image-based processing to perform object tracking, detection and monitoring [79].

Traditionally, the video-based approach was used for HAR. However, recent studies explored still image-based action recognition. The video-based approach performs complex methods to process and compare sequential images along with motions to track individual objects. Therefore, many Spatio-temporal features and methods introduced for traditional video-based approach do not apply to image base processing still.

Figure 2.13. Illustrating the challenge of recognising and describing a semantically equivalent "kicking" action

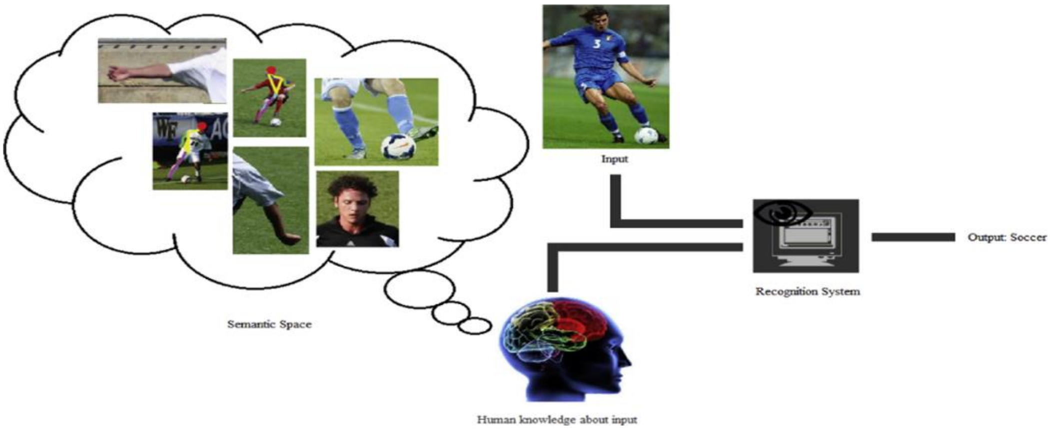


Figure 2.14. Depicting the semantic space concept using pose, poselet, object/scene, context, and attributes methods

In the recent literature review presented in [79], focuses on the recent semantical image processing frameworks. It presents the challenges and methods adopted by the researchers to recognise a single activity from an image which can be performed by any human being with different body posture, size and shape, clothes, camera angle, etc. For instance, Figure 2.13 illustrates a kicking action performed in different ways, camera angle, location and body posture styles, but they are semantically the same. It has been argued that integration of semantical knowledge-based approach can describe the characteristics of the complex situation activities and the non-semantical approaches are only ideal for describing simple actions. They introduced the concept of semantic space which comprises of pose (whole human body), poselet (individual body part), object/scene, context, and attributes methods. Figure 2.14 depicts the concept of semantic space from a given image as an input (1), recognition system (2) which decomposes the image with above methods and uses human knowledge models to infer the activity being carried out and produce it as an output (4). In addition, linguistic descriptors and reasoning-based hierarchical semantic representation have been used for semantically extracting meaningful results. It was highlighted that low-level features do not always provide the highest data extraction due to the significant variation in scales, viewpoint, and pose in real-world data. Finally, it presents the system applications such as recognising untrained or unknown activity using human knowledge as a zero-shot learning method, early AR, gapped-video (missing image frames), activity forecasting and analysis.

There are various algorithms available to process these still images. The popular ones are Scale Invariant Feature Transform (SIFT), which is designed to detect features more accurate and in contrast Speed Up Robust Feature (SURF) which aims to perform faster but may lose accuracy[80]–[82]. In additions, many variances of theses algorithms also exist, and they all have their strengths and weakness.

This image analysing capabilities opens up many opportunities for various domains such as facial [83]–[85], and abnormal activity detection[86] for security, augmenting surgical instructions in the medical field and automotive industry for self-driving cars.

2.3.1.2. Sensor-based approach

On the other hand, a sensor-based approach uses diverse types of sensors which can be worn (wearable sensors), embedded in everyday objects (dense sensors) and distributed in the environment (ambient sensors). Firstly, the wearable sensors are the devices which can be directly or indirectly worn by the user such as smartwatch and head mount displays, i.e. Google Glass and Oculus Rift. These devices can have their own set of sensors which can stream the contextual and physical movement data of the user for inferring various activities, i.e. Smart-Cuff[87]. Secondly, dense based sensing is when the sensors such as contact sensors and accelerometers are embedded into our daily objects such, i.e. kettle and teapot. These types of sensors generally provide binary represented of the state of the devices, i.e. on or off. Finally, the ambient sensing enables monitoring of an environment using interconnected sensors that un-obstructively such as motion sensors, temperature sensors, and radio frequency identification (RFID) tags. presents an overview of diverse sensing technologies in various categories, along with some of the popular devices used in different AR studies.

The significant advancements in hardware capabilities have given us smaller and cheaper devices which can be used for many ways. In the dense sensing, the microelectromechanical systems (MEMS) chips are commonly used due to its size (thick as a human hair). More recently, these MEMS devices along with other sensors are getting integrated into objects such as golf bat, tennis rackets, boxing gloves, soccer, tennis racket, baseball, softballs, running shorts, basketball, and helmets to allow one to analyse their performances, detect, monitor and forecast potential problems[88].

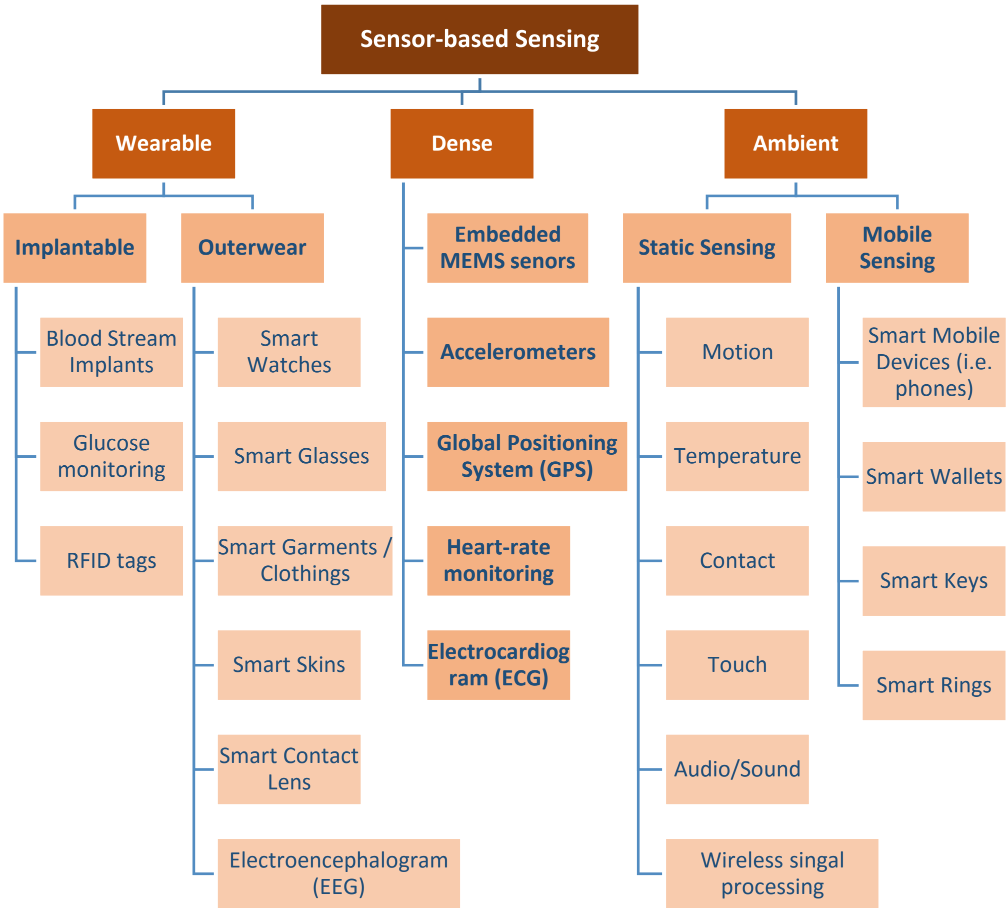


Figure 2.15. Categorising of sensor-based sensing technology

The wearable sensing devices can be further categorised as implantable and outerwear. The outerwear wearable sensors devices can be smartwatches, head mount displays (i.e. virtual reality glass and Google glass), and Shimmer sensing platform[19] has been used in several AR-related studies. The wearable clothes/smart garments comprehensively listed here [89] can be further utilised for fine-grained action monitoring and recognition. Some of the smart clothing project listed are Heddoko, Hexoskin, Ralph Lauren polo tech shirt, SMOOZI D-Shirt, OMSignal, Athos, Gymi smart shirt, AIQ smart clothing, and Mimo baby monitoring. Also, the development in haptic-based technologies[90] can be used for not only for human-computer interface (HCI) but also collecting, monitoring and inferring user's behavioural state[91]. In contrast, there are also smart devices which can be implanted into the human body for various purposes such as health monitoring and regulating (i.e. glucose level[92]), and human identification and positioning (i.e. implanted RFID-tags [93]–[95]).

The ambient sensing technologies have been widely adopted to collect data and monitoring ambient users in a non-intrusive, non-invasive and ubiquitous manner. The ambient sensing devices could be further classified in static and mobile sensing due to nature. The static devices such as motion, temperature, door sensors, RFID readers, wireless signal process, and audio processing devices are typically positioned in a single given location and are unlikely to be moved frequently. Whereas, devices such as smart phones[96], [97], smart wallets[98], smart keys, and smart rings can be classed as mobile sensing. These mobile sensing devices do not have fixed location and highly likely to be used for collection fine-grained activity monitoring and recognition.

One of the recent work [15], employed ambient sensing devices; passive infrared (PIR) sensors for motion detection, room temperature and light data to infer set of ADL. The study presented a novel clustering framework to pre-process the raw data from the above ambient sensors before performing the actual classifications of the ADLs by using the three state-of-the-art classifiers (NB, SVM and random forest (RF)). Figure 2.16 illustrates the phase where a new clustering algorithm is applied. A dataset was collected for over 200 days to evaluate and compare the performance of the three existing classifications. The result indicated that RF classifier outperformed NB and SVM with the following factors: an average specificity of 96.53%, a sensitivity of 68.49%, a precision of 74.41% and an F-measure of 71.33%. However, some of limited of this system is that it requires manual work on the preparation of data collections, i.e. several sensors, sampling rate and data formatting. Besides, it only supports single-person in multi-rooms to recognise single activity as the segmentation process will not work for multiple or interleaving actions occurring in the same room.

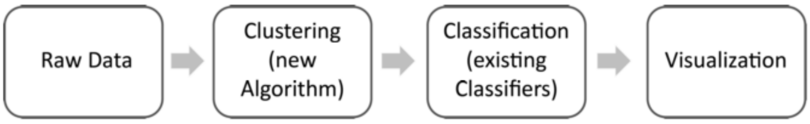


Figure 2.16. The novel clustering algorithm applied to pre-process raw sensing data before performing classifications and visualisation.

2.3.1.3. Multimodal/Fusion

The multimodal approach combines the vision and sensor-based sensing capabilities, where the motives can be defined as to achieve a balance on the level of computation, complexity, privacy and accuracy of the data depending on the overall requirements of the system. For instance, the work in [99] proposes an ontological framework to perform activity recognition in a smart home environment. The framework fuses sensor-based and vision-based techniques to collect the data from the environment and uses an ontology to describe the relationships and interactions between various activities, monitoring entities, user and the sensory data. Another work in[50], presents an energy-efficient mobile sensing system (EEMSS) where the combination of GPS, WIFI, accelerometer, and microphone components are used to detect various user states. These activities are "*Walking*", "*Vehicle*", "*Resting*", "*Home talking*", "*Home entertaining*", "*Working*", "*Meeting*", "*Loud office*", "*Quiet place*", "*Speech place*", and "*Loud place*".

2.3.2. Smart Home Environments

SH environment is created with heterogeneous sensing technologies commercially available and by developing bespoke sensors with the support of microcontrollers. These sensing technologies are categorised as vision and sensor-based approaches as detailed in section 2.3.1.

Several bespoke low-cost microcontroller-based sensing solutions have been proposed with wireless connectivity, cloud platform and remote access to SH devices[100]–[103]. Work in [101], presents Frugal Labs IoT Platform (FLIP) to monitor and control the SH environment. It leverages microcontroller and MQTT cloud platform for SH monitoring, control (appliances/lighting), detect (intrusion/smoke/gas) and alerting (danger/anomalies). Similarly, a miniature microcontroller such as ESP8266 nodeMCU microcontroller with firebase cloud platform is leverage [103] to control electrical appliances in SH environment using Android application. Another work [104] recommend recipes to cook based on objects and ingredients available in the kitchen. The objects and ingredients are attached with an RFID tag and read by the reader connected to a microcontroller. Other microcontroller-based solutions are also proposed for controlling and viewing their status only [102], [105]–[107] without any cloud platform for performing any analysis or automation tasks.

Although microcontroller-based solutions are flexible, reduce cost and has higher scalability, there are several challenges with this approach. One of the critical limitations of this approach is that it requires expert knowledge to set up the system and when adding new sensors. The setup process involves three main steps, (a) wiring sensors to the microcontroller, (b) programming microcontroller and (c) software system collecting data. Hence, each time a new sensor needs to be added, the three-step setup process needs to be repeated. There have been some efforts being made to ease the three-steps setup process for microcontroller-based solutions such as over-the-air (OTA) programming/firmware upgrade. However, it remains a challenge to create "plug-and-play" solutions.

The commercial smart home kits are now emerging with proprietary and open source components. These kits contain a variety of devices for vision and ambient sensing (i.e., temperature, lighting, switches, motion, and door/window) technologies. For instance, SmartThings and Almond Guard are some of the popular kits available over Amazon. Another more specialist security kit such as Arlo has advanced features such as intrusion detection, monitoring and alerting. Generally, these kits come with a multiprotocol smart hub that supports wired and wireless sensors, i.e., ZigBee, Z-Wave, and WiFi. For instance, Almond Guard and SmartThings both supports wireless sensors from a variety of manufacturers and can connect new devices effortlessly. Despite the ease of fitting wireless sensors in the desired location, the wireless sensors have limited battery source and require frequent replacement. Other individual devices connected to main power lines are also available that can control lighting and electrical appliances such as Philips Hue, TP-Link and WeMo switches and plugs. With the vast diversity in sensing technologies and manufacturers, the complexity of interacting with all of the devices with the individual mobile application one of the key technical challenge. New waves are now emerging, such as Amazon Alexa and Google Home that can interact with smart sensors within SH environment with voice-commands. The speech-based human-machine interaction is advantageous for non-technical expert users to naturally interact with the system.

2.3.3. Emerging AAL Platforms

Several SH based AAL systems have recently emerged from the European Commission and other institutes/privately funded projects around the world. Work in [108] present guidelines and recommendations from the experiences of developing the smart living space with ambient intelligence at the University of Jaen (UJAmI). UJAmI was designed with AAL and monitoring activities and behaviour in mind. UJAmI is equipped with many sensing modalities. UJAmI currently has two publicly available datasets containing vision- and sensor-based data. One of the limitations of a sensor-based dataset is the lack of diversity in sensors attached to everyday objects as it assumes interaction with binary sensors and their presence with proximity (BLE tags), motion or floor sensors. Additionally, vision-based dataset currently contains one scene with several still images. Although datasets like the opportunity [109], contains ambient, embedded and wearable sensors are useful, there are still many practical challenges such as wearing multiple sensors around their body everyday daily life with limited energy for continuous monitoring of body movements. An alternative method could be to embedded the inertial sensors within the everyday objects with on interaction/near proximity-based data exchange features to save energy and making wearing sensors optional.

The middleware referred to as SensorCentral is proposed by Smart Environment Research Group (SERG) from University of Ulster [110]. The key motivations of the platform are to handle big heterogeneous data storage with several research-oriented features. Some of these features include data annotation, metric generation, exporting experimental datasets, machine learning services, rule-based classification, broadcasting live sensor streams and quick sensor configuration. Moreover, SensorCentral is motivated to support the Open Data Initiative

(ODI) so that dataset collected can be shared and re-evaluated by others in a common framework.

A general-purpose Human Health and Activity Laboratory (H2AI) research facility developed at Lulea University [111]. H2AI emerged as a collaborative effort mainly from three European universities: Jaen (UJAmI), Halmstad (HINT) and Ulster (SERG) universities. H2AI consist of wide varieties of wearable sensors, cameras, and ambient sensing platforms. H2AI has integrated its existing iMotion platform with SensorCentral. H2AI is also looking to incorporate other platforms developed by industries (i.e., Tieto SmartCare platform and eSense) and other projects (i.e., FIRWARE).

2.4. Activity Recognition (AR)

The development of activity recognition (AR) approaches is influenced by the KD and DD activity modelling approaches leveraged. However, before analysing any data collected from the smart environment, data need to be segmented into relevant ongoing ADLs. The accuracy and speed required for segmenting observed actions within a single or mixed activities scenario are essential. In addition, users in real-world are likely to have personal preferences on conducting ADLs with unique ingredients or a variety of utensils. Hence, creating a challenge to not only model generic and user-preferences but also incorporating the model in the data segmentation process. In-depth literature review challenges in developing a data segmentation approach is presented in CHAPTER 3.

The data processing and pattern recognition phase in the AR process brings together the models created before the system execution and expert training data or explicit knowledge to detect various activities and adding the labels on them. These labelled activities can then be used to match against the pre-defined user preferences or activity model to identify if the user requires further assistance with the activity. The approach such as sliding window protocol, [20], [24], [25], [112]–[114], is used to create a window either statically or dynamically depending on the nature of the activity to monitor over a particular duration of time.

The work in [25], presents KD approach for Concurrent Activity Recognition (KCAR) that is being performed by multi-users. KCAR segments continuous sensor events into fragments by inspecting individual sensor events and finding the semantic similarity of the activities specified in the ontological models. These fragments are continuously evaluated to determine if the activity is completed, on-going or out-of-date (pre-defined maximum time gap). This approach can distinguish activities at the course-grained level by using attributes of the events such as location and object properties, i.e. cook and work activity occurring concurrently. However, it is unable to detect information such as how many people are involved in performing the activities and which user is performing a particular action. Moreover, fine-grained levels of activities are also not detectable; activities such as retrieving ingredients, hand washing dishes and stirring a pan while cooking.

Sensor similarity

$Sim(se_i, se_j) = (Sim_{Time}(se_i, se_j), Sim_{Sensor}(se_i, se_j))$

$Sim_{Sensor}(se_i, se_j) = sim_{Conceptual}(location(s_i), location(s_j)) + sim_{Conceptual}(object(s_i), object(s_j)) / 2$

Conceptual similarity function

$$Sim_{Conceptual}(c_1, c_2) = \frac{2 \times N_3}{N_1 + N_2 + 2 \times N_3}$$

N_1 and N_2 is the count of many nodes upward till parent node of two concepts (Least Common Subsumer (LCS)) and N_3 is the number of nodes from the parent node (LCS) till the root node. Figure 2.17 illustrates two examples of conceptual similarity function equation in action between Stove and Fridge; and Stove and Computer with the given hierarchical environmental object description in the ontology.

Time similarity function

$$Sim_{Time}(se_i, se_j) = \max\left(0, 1 - \frac{|t(se_j) - t(se_i)|}{T_{max}}\right)$$

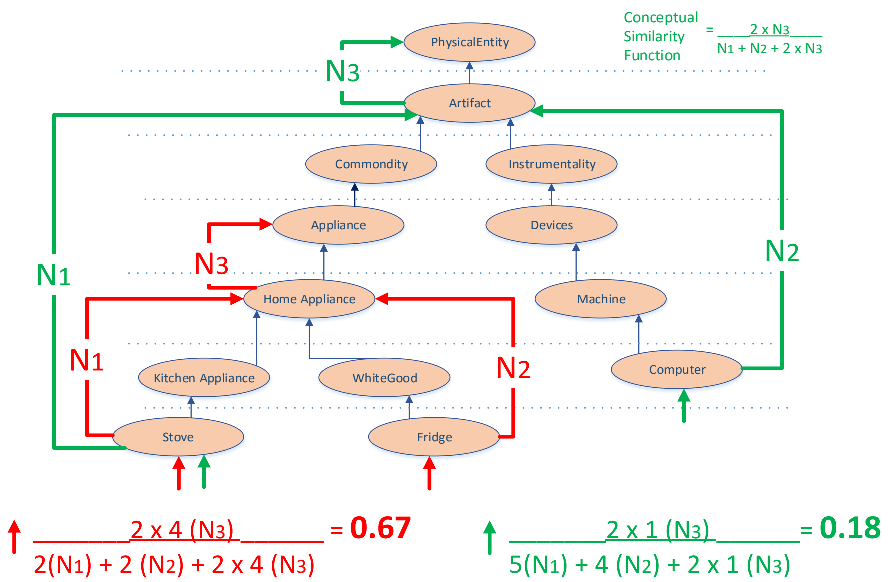


Figure 2.17. Illustrating the sensor and ontology modelling similarity approach

Furthermore, to deal with occasional sensor noise in the input sequence, it has been said that pure knowledge-based reasoning approach falls short and may reach contradictory conclusions. Therefore, work in [25] adopts Pyramid Matching Kernel (PMK) approach, to perform approximate matching on sensor event sequences and activity profile. PMK technique is based on image-based object detection where the images are compared by segmenting the image into grids at different levels to detect key features and perform statistical, which also has the capability to perform matching on hierarchical concepts.

The work in [78], presents a hierarchical nonparametric modelling approach based on the distance-dependent Chinese restaurant process (ddCRP) to infer activity routines from the sensor data streams. ddCRP does not require any labelled data or depend on time-invariant sliding windows. The authors' segmented context words into supersamples using context state and formulated a segmentation prior with semantic and temporal information to group supersamples that belong to individual activities using ddCRP and CRP.

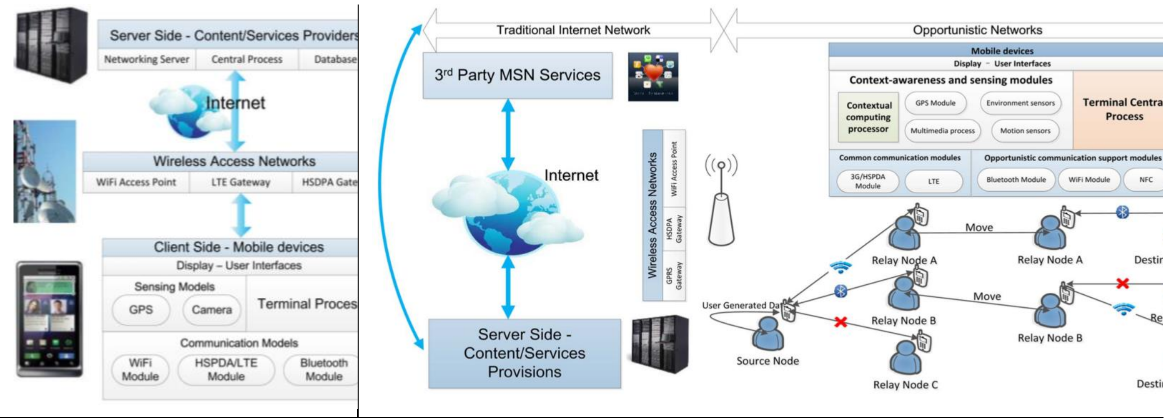
The work in [115], proposes to combine contextual attributes (the object used, time and location) and fixed time interval for the sensor segmentation process. The fixed time interval will shift only if the occupancy duration in an area is below a minimum or maximum threshold. The generated event sequences are then passed to the activity recognition (AR) algorithm to identify both simple and mixed activities. AR is performed using Markov Logic Network (MLN) approach, which combines probabilistic and logical reasoning in a single framework to represent uncertainty and domain knowledge.

The inferencing and reasoning process occurs after separating and segmenting process. Different classifiers in discussed for the DD and KD activities are adapted. In the case of the DD approach, different algorithms are developed upon the nature of the use case scenarios and the types of data being collected. These algorithms can also be used on top of KD ontological models to perform inferencing and reasoning. Due to the semantic nature of the knowledge-driven approach, formal reasoning engines [116] can be used to deduce additional facts from a given set of rules autonomously.

2.5. System Architecture for Ambient Assisted Living (AAL)

The work in [77], presents a layered system architecture for the IoT and smart home systems using IoT along with their problems and challenges that are currently being faced with its growing applications. Some of the challenges and difficulties recognised are that there is no standardisation of the system in terms of security (device identification, authentication and communication protocols), integration, coordination, data storage and mining, and self-organising devices and network structure.

The survey in [117], provides a comprehensive overview of mobile social networking (MSN) in terms of their applications, platforms, system architecture and highlights some future research directions. MSN is a social networking medium where individuals with similar interests interact with each other through their mobile devices, i.e. commercial platforms such as Facebook, Twitter and Foursquare. MSN leverages on mobile communications networks and hardware components of the mobile devices to have ubiquitous accessibility. It compares MSN with traditional social networking and highlights in many ways in which mobile devices can be used. In their work, they review a variety of MSN platforms, existing systems solutions and proposes an overall architectural design for conventional and future MSNs systems. These two system architectures adapt client-server style, i.e. service-orientated architecture (SOA). Also, different stakeholders' perspectives views are taken into consideration: physical (system engineers), development (application developers) and logical views (end users).



a) The physical architecture of the conventional MSNs

b) The physical architecture of the future MSNs

Figure 2.18. The physical system architecture of the conventional and future MSNs

Additionally, many social behaviour analysis applications and context-awareness applications were reviewed in terms of the stakeholder's perspectives mentioned above, their current features and potential needs in future. Figure 2.18 provides an overview of the SOA style physical architecture of conventional MSNs (a) and future MSNs (b). In the physical architecture style of the conventional MSNs, the standard components of server-side and client-side (mobile devices) are used via wireless access networks (WANs). For instance, the sensing and communication modules on the mobile devices to communicate the data wirelessly with the server to store and process the data using any given wireless protocol. On the other hand, the future MSNs physical architecture extends the approach by adopting a hybrid approach to the architecture system from every aspect such as style, network communication, data flow and storage and connectivity. For instance, it does not only adapt client-server style but also peer to peer, where nearby devices can use short-range communication protocol such as Bluetooth, infrared and ad-hoc wireless connection. Figure 2.19 presents the full comparisons of the two architectures discussed above.

	Conventional MSNs	Future MSNs
Architecture	client-server	hybrid (client-server + peer to peer)
Network	Internet	hybrid (Internet + opportunistic networks)
Data Flow	end-to-end	end-to-end + opportunistic contacts
Connectivity	rely on 2G,3G,4G & Wi-Fi Hotspot	make use of any connectivity available
Data Delivery	Ensure quality of service	Ensure quality of service in Internet and provide best effort services for delivery in opportunistic networks
Data Storage	mainly on server side	on both client and server sides
Features & Services	1. communication interacting 2. updating personal status 3. advertisement 4. location service	Include all features and services in conventional MSNs and 1. message/file transfer 2. media streaming 3. content dissemination 4. neighbor discovery
Key technologies	1. social behavior analysis 2. context-aware	Include all key technologies in conventional MSNs and 1. ad-hoc connectivity 2. data forwarding & data dissemination 3. simulation of human mobility pattern
Application Domains	1. social networking services 2. game 3. travel 4. business 5. education 6. healthcare 7. dating 8. road traffic	Provide a more comprehensive coverage in the application domains of conventional MSNs and include new application domains such as 1. social networking in developing/ rural regions 2. disaster relief 3. vehicular networks

Figure 2.19. Key feature comparisons between conventional and future MSNs

The future architecture style was applied in the vehicular social networks (VANs) systems case study to create a vehicular ad-hoc network (VANET). This architecture style was able to bring together vehicle-to-vehicle (V2V), vehicle-to-pedestrian (V2P), and vehicle-to-roadside (V2R) infrastructure to collaboratively work together ubiquitously. However, there are many technical challenges highlighted such as privacy and security, resource management and user behaviours, efficient data mining approaches, efficient energy management, and efficient integration of various IoT devices. Moreover, some future research directions were presented, such as RESTful web service and cloud computing and mobile crowdsourcing that has the potential to change the landscape of many other application domains.

In work [118], the survey of RDF data store solutions is presented, which allow the semantical data to be stored, queried and manipulated efficiently. The RDF store, also known as triplestore, is a dedicated software component that is built on top of the traditional data management system. The survey highlights that the RDF store is suitable to be used as the backbone of any semantic applications that need to store and process a large amount of data safely and reliably. However, some limitations were emphasised such as no dynamic inferencing on the existing RDF data store, performance benchmarking of the RDF stores and versioning/handling the streaming data in an efficient way (from an industrial point of view to trace back and identify problems).

2.6. Overview of HAR

Table 2.10 presents an overview of state-of-the-art KD and DD approaches adapted for HAR. In summary, the DD approach was initially adapted to address activity recognition problem. The data-driven approaches require pre-collected data to be analysed to create the activity model. The methods adopted to process the datasets can be categorised in the generative and discriminative method. The generative method aims to map all the sensor data concerning their class and create a probabilistic model. The standard classifiers adapted for generative approaches are NB, HMM, LDS and DBNs. The discriminative methods are used to further classify from the generic model using similarities or rule-based reasoning. The conventional discriminative classifiers are NN, SVM, CRF and decision tree. In general, the data-driven approach requires pre-processing large datasets, creating a cold start problem and has poor reusability and scalability problem. However, this approach handles modelling uncertainty, temporal information and ideal for evolutionary learning.

Table 2.10. Comparisons table of data-driven and knowledge-driven approach [36]

	Knowledge-Driven Approaches (KDA)				Data-Driven Approaches (DDA)	
	Mining-based	Logic-based	Ontology-based		Generative	Discriminative
Model Type	HMM, DBN, SVM, CRF, NN	Logical formula, e.g., plans, lattices, event trees	HMM, DBN, SVM, CRF, NN	Sensor and Activity ontologies	Naïve Bayes, HMM, LDS, DBNs	NN, SVM, CRF Decision tree
Modeling Mechanism	Information retrieval and analysis	Formal knowledge modeling	(un)supervised learning from datasets	Ontological engineering	(un)supervised learning from datasets	(un)supervised learning from datasets
Activity Recognition Method	Generative or discriminative methods	Logical inference, e.g., deduction, induction	Generative or discriminative methods	Semantic reasoning, e.g., subsumption, consistency	Probabilistic classification	Similarity or rule based reasoning
Advantage	No “Cold Start” problem, Using multiple data sources	No “Cold Start” problem, clear semantics on modeling & inference	Shared terms, interoperability and reusability	No “Cold Start” problem, multiple models, clear semantics on modeling & inference, interoperability & reusability	Modeling uncertainty, temporal information	Modeling uncertainty, temporal information, Heuristics
Disadvantage	The same problems as DDA	Weak in handling uncertainty and scalability	The same problems as DDA	Weak in handling uncertainty and time	“Cold start” problems, Lack of reusability & scalability	“Cold start” Problems, Lack of reusability & scalability

KD approach, on the other hand, benefit from the fact that human knowledge is conceptualised formally. KD approaches can be subcategorised as mining, logic and ontology-based. The mining-based approach utilise already existing dataset available online to create models which can be classified in a generative or discriminative manner; similar to the DD approach. The logical based approach uses mathematical foundation to formally represent knowledge and implement rules using various activity theories to perform logical inferencing and reasoning. In general, the knowledge-driven approach suffers from handling uncertainty and scalability issues. The ontology-based model represents knowledge in triplet format with rich semantics, enabling reasoning to infer and deduce additional facts which may not be explicitly defined. However, addressing cold start problem at the start, reusability, and implementing rich relations semantically in the datasets which can be shared across domains.

The hybrid approach was later introduced to combine the capabilities of DD and KD approaches to address their limitations. KD approach is used for modelling the activities, adding semantics and inferencing various activities and the DD approach for activity learning (using un-/supervised) and evolve KD model from the observed dataset[20].

Table 2.11 provides an overview of the recent studies carried in addressing the crucial challenges in mixed activity recognition. The key areas and problems in AR, strategies and technologies adapted and some of the shortfalls have been highlighted.

Table 2.11. Overview of the recent studies carried out in human activity recognition

Application/ Framework	**Challenges addressed**	**Single/ Multi-users**	**Sensing Type(s)**	**Modelling method(s)**	**Personalised/ Generic**	**Notes/ Comments**
			Data-driven (DD)			
HHMM-JPDAF [23]	Mixed	M	Vision	HHMM, JPDAF, RBPF	Both	Behaviour analysis and tracking
SACAAR [46]	Mixed	S	Sensor-based	Probabilistic and Markov chain analysis	Both	Context-driven activity theory (CDAT), Decision Trees (DT) (J48) and Naive Bayes (NB)
Energy-Efficient Mobile Sensing System (EEMSS) [50]	Single	S	Sensor-based	Decision tree (DT)	Generic	Energy-efficient mobile-based AR method. Require offline training models for DT.
			Knowledge-driven (KD)			
KCAR [25]	Mixed	M	Multimodal	Ontology modelling	Generic	Dynamic sliding window, Pyramid Match Kernel (PMK)
Necesity [119]	Non-/ Sequential	S	Sensor based	Ontology, Rule based	Generic	SWRL
Ontological framework [99]	Non-/ Sequential	S, single activity.	Multimodal	Ontology	Generic	Combining ontologies, sensor and video data.
Dynamic segmentation [27]	Non-/ Sequential	S	Sensor-based	Ontology	Generic	Dynamic sliding window
MetaQ [120]	Non-/ Sequential	S	Multimodal	Ontology/ Query	Both	SPARQL based
Recognising Activities of Daily Living (RADL) [121]	Non-/ Sequential	S	Sensor-based	Ontology	Generic	Jess + Protégé
Ontological and temporal formalisms [24]	Mixed	S	Multimodal	Ontology, Rule-based	Generic	4D fluents, OWL DL, Allen temporal logic, SWRL, Jess + Protégé

2.7. Challenges and Opportunities within HAR

HAR capabilities within a smart environment pose many challenges at all the five phases defined in section 2.1.2. Firstly, the data collection phase consists of employing diverse sensing methods which has several challenges and open issues that need to be addressed such as privacy, security, practical, interoperability, technical challenges and financial implications. Section 2.7.1 elaborates on some of the challenges to monitor and capture the inhabitants' behaviours using available sensing techniques. Secondly, the data segmentation phase requires sensor events to be disentangled into a relevant set of ongoing ADLs despite the order of actions and mixed activities; more details are provided in section 2.7.2.

Thirdly, data modelling phase has several challenges to address, such as intricately describing ADLs, complex smart sensing infrastructure, environmental entities, and user profiles for data segmentation and activity recognition (fourth) phase. Further challenges are created when modelling imprecise non-binary sensor data and several uncertainties caused when performing mixed activity recognition accurately. Section 2.7.3 expands on problems related to mixed activity recognition while sections 2.7.4 and 2.7.5 discuss challenges in modelling fine-grained user actions with imprecise sensors and uncertainty factors influencing the activity recognition accuracy.

Fourthly, the activity recognition phase is influenced by the KD or DD activity modelling approaches selected to recognise single and multi-user activities. Section 2.7.6 elaborate on challenges of detecting, identifying, and associating user actions in the shared living environment. Lastly, the activity learning phase needs large datasets that are well-formatted and annotated to perform pattern, and frequency algorithms developed using supervised, or unsupervised DD approaches. Nevertheless, investigating on challenges related to activity learning phase is out of the scope for the thesis as discussed in section 1.4.

For each of the abovementioned AR phases, various interdependent underlying technologies also exist [36]. These technologies present further system architectural integration challenges, which are mainly due to their differences in programming languages, development environments, proprietary components, data storage and communication protocols. Therefore, the interconnectivity of each phase into a single platform poses several challenges when developing a unified system architecture styles and patterns tailored to the AAL system. These AAL system architectural level and big data storage challenges are elaborated in sections 2.7.7 and 2.7.8.

2.7.1. Multimodal Smart Sensing Environment

The vision-based approach has been used extensively for security and surveillance systems. However, concerns with the privacy of inhabitants in their private homes have led researchers to explore unobtrusive and pervasive sensor-based approaches.

Another challenge is that several vendors make application-specific off-the-shelf products that are not always open-source and run on diverse communication protocols. Hence, creating a big problem when integrating these cross-manufacturer devices within WSNs of any given size. However, to address this challenge, many efforts have been exerted by the vendors in recent years. One common practice is to provide application program interfaces (APIs) and software development kits (SDKs) to allow cross-platform third-party service integrations. For instance, Securifi Almond+ router, Amazon Echo [122], and Samsung SmartThings [123] can interact with each other's devices. Although these services are growing, limited intelligence can be added to the sensor nodes as rules govern them, such as "*if this, then that*" concepts (i.e., IFTTT [124]). Furthermore, they still have limited types of sensors that can support fine-grained sensing capabilities for AR, i.e., a capacitive touch sensor on an object for dense sensing. Therefore, bespoke Arduino-based wireless sensing methods are still commonly used [125], [126]. Therefore, this thesis investigates on the challenges to integrate some of those above off-the-shelf and open-source WSN technologies within the AAL system architecture to achieve real-time AR, monitoring, and assistance provisioning.

2.7.2. Sensor Data Segmentation

One of the critical challenges for segmenting data continuous sensor data stream is to disentangled single and mixed activities with actions conducted in any order. Also, most studies model generic set of actions for ADLs when segmenting continuous data and performing AR. However, users in real-world are likely to have personal preferences on conducting ADLs with unique ingredients or a variety of utensils. Hence, creating a challenge to not only model generic and user-preferences but also incorporating the model in the data segmentation process.

2.7.3. Mixed Activity Recognition (AR)

Activity recognition (AR) phase rely on the segmented sensor data and ADL knowledge model to recognise single and mixed user activities. Developing the ADL model with KD approach need domain expert knowledge and manual effort to maintain/evolve the model compared to the DD approach requiring large pre-collected dataset to train the model. KD model is reusable with other users and can be shared across domain is contrary to the model developed with DD. Both approaches can only define a finite set of activities in the model. Hence, the hybrid approach must use the initial model developed in KD and evolve with the DD approach.

2.7.4. Fine-grained Action Detection with Multimodal Sensor Data

Recent studies recognise activities by assuming completion of an action using binary sensors. However, more attention is required to fuse multiple sensing attributes of an object to recognise user interactions as fine-grained level. However, non-binary sensors data from the smart environment create impreciseness and subjective interpretation for the status of the object, i.e., if the cup is "*low*" or "*half full*". Hence, the fine-grained action modelling approach is required for ADLs to not only define imprecise concepts but also fuse sensing attributes.

2.7.5. Uncertainty Factors in AR

Several problems related to technology failure, human error, environmental condition and object functionality can affect the reliability and trustworthiness of the observed sensor data and AR results. For instance, a low battery level of a sensor can impact the sensor reading accuracy and transmitting range due to a weak signal, which can ultimately cause error, delay or loss of packets. Thus, influencing the AR results. Hence, uncertainty theories such as probabilistic [127], [128], evidential [129], [130], and fuzzy [57], [131] need to be investigated to model and reason with the uncertainty factors present in AR.

2.7.6. Multi-user AR in Shared Living Environment

AAL systems are likely to be deployed in a dwelling with more than one occupant. Hence, some of the main challenges are to detect if there are multiple occupants in the same environment, identify the occupant using discriminating sensing techniques and associate their actions with ongoing ADLs[28]. Therefore, a multi-user AR approach is needed with a smart environment equipped with non-obstructive sensing approaches that preserve the privacy of the occupant and security of the personal data.

2.7.7. AAL System Architecture Style and Patterns

One of the main challenges in building an assistive system is to select appropriate system architectural styles and patterns which can be easily misused [132]–[134]. Engaging with the broader community by having open source components and using popular programming languages can play a crucial role in coming up with useful, adaptive, and personalised solutions. Other factors influencing the design decisions include semantical data storage, computation power requirement, low latency communication protocols, and the ability to allow simultaneous access to the users with a convenient human-computer interface (HCI). Some of the exiting assistive systems (explored further in section 8.2) are built in a standalone application environment. However, questions have been raised regarding its extensibility, reusability, scalability, maintainability, and/or use of proprietary components, which may have limited community support. In addition, having a poor or an unnatural HCI design poses practical limitations for its key users.

Over the years, the service-orientated architecture (SOA) approach has become popular, because it can address some of the aforementioned issues as well as create a mechanism by which to delegate resource-intensive tasks and storage to powerful sets of computers over a network (cloud computing). Moreover, using the SOA approach also allows low-power devices such as mobile devices or any other gadgets with network capabilities, to utilise the available services. This has not only improved the HCI of the AAL system but also made it scalable such that it can serve cross-platform clients as well as integrate and reuse third-party services in a creative manner. SOA approach now drives the concepts of SH, IoT, and ubiquitous or pervasive computing. This is the main approach by which everyday objects can be seamlessly integrated into the interconnected World Wide Web (WWW).

2.7.8. Semantic Data Storage

Another challenge faced that arises from this topic are the problem of storing the activity modelling and recognition data using the semantical structure that can be used in a meaningful way. The storage options considered here also influence the overall system architectural design decisions. Recently, this has become a much broader issue with the accumulation of large amounts of unstructured or in semi-structured data with no clear semantical relations. Hence, creating many problems such as automating the task of processing and retrieving data efficiently [70]. Currently, machine-learning techniques, such as genetic algorithms are used to extract and train computers on how to process the data over time. This approach, however, is lengthy and needs a high computation rate.

For efficiency, the concept of the semantic web was introduced. This concept was initially envisioned by Tim Berners-Lee and his colleagues for creating the WWW with linked data structure with semantic meanings defined using formal methods that can be processed by a machine [71], [135]. Hence, representing the data in the form of a triplet, subject-predict-object to form a connected graph. The most common vocabularies are used and shared to create an expressivity in the data (i.e., using Resource Description Framework (RDF) [136], [137] and Web Ontology Language (OWL) [138]). Also, various reasoning engines (i.e., Pellet, HermiT, and FaCT++) are used to perform inferencing utilising the user-specific rules and formal languages. The set of triplets are stored in the triplestore (database) as a graph, which is specially optimised for handling them. Moreover, just like the Structured Query Language (SQL) in traditional relational databases, the SPARQL Protocol and RDF Query language (SPARQL) are used to perform, create, read, update and delete (CRUD) operations [137], [139].

These capabilities and benefits enable the back end of any application to achieve greater flexibility within its specific system architecture.

2.8. Summary

This chapter presents a research background of HAR with five key phases, reviews of state-of-the-art studies related to these phases, and highlights of critical challenges, opportunities and open issues identified as a result. The complementary relationship between the five HAR phases to develop a suitable system architecture based AAL system requirements and integration of multimodal smart environments are discussed.

The following chapters will investigate on eight critical challenges identified in section 2.7, propose and evaluate novel methods, approach and framework. CHAPTER 3 examines on data segmentation challenges described in sections 2.7.1 and 2.7.2 to utilise semantical relationships of the sensors to associate a set of actions to a given ADL. CHAPTER 4 builds on the challenges of recognising mixed activities at multi-granular levels by fusing multimodal sensors data and ADLs model as described in sections 2.7.1, 2.7.3 and 2.7.4. CHAPTER 5 investigates on the uncertainty factors that exist in recognising ADLs accurately as described in section 2.7.5. CHAPTER 6 presents an overall framework for single AR, and CHAPTER 7 extends to identify and associate multi-user actions within a shared smart environment, as highlighted in section 2.7.6. CHAPTER 8 analyses the state-of-art system architecture for AAL systems and proposed a suitable microservices architecture with graph-based big data storage requirements discussed in sections 2.7.7 and 2.7.8. Finally, CHAPTER 9 summaries each chapter, presents key contributions made in this thesis, discuss challenges and open issues to be addressed in future work and offer concluding remarks.

CHAPTER 3. SEMANTIC-ENABLED SENSOR DATA SEGMENTATION

The first challenge in the activity recognition (AR) process is the segmentation of observed sensor events into an ongoing set of activities of daily living (ADL). Several studies have proposed approaches to unravelling and organising sensor observations to support the recognition of generic ADLs performed in a sequential or mixed activities scenario. However, not enough is explored in semantically distinguishing individual sensor events directly with the knowledge of user preferences and passing it to the relevant ongoing activities. Hence, this chapter proposes the Semiotic theory inspired the ontological model for capturing generic and user-specific ADL preference knowledge to support the segmentation process. Subsequently, this chapter proposes a multithreaded decision algorithm to segment continues sensor data stream into a set of ADLs and provide system implementation details on tools and techniques applied. This system prototype was evaluated against 30 use cases including sequential and mixed activities scenarios where each event was simulated at the 10-second interval on a machine with dual-core i7 2.60GHz CPU and 8GB RAM. The result illustrates that sensor observations were segmented with 100% accuracy for single ADL scenarios and 97.8% accuracy for mixed activities scenario. Nevertheless, the performance has suffered to segment each event with the average classification time of 3971ms and 62183ms for single and mixed activities scenarios, respectively. This chapter concludes with a summary and discusses future work based on the shortfalls and optimisation opportunities of the system.

3.1. Introduction

Human being conducts one or more activities in a sequential, interwoven and concurrent manner as discussed in section 2.1.1. The actions for individual activities of daily living (ADL) are performed in any order and assumed to be independent of previous activity. However, some dependencies between user actions for a given activity or shared action between two or more activities can exist such as opening a fridge to take milk and margarine out at the same when preparing to make tea and toast for breakfast. Therefore, the observed sensor data from the smart environments produce entangled set of user actions over a given time interval which makes it challenging to perform activity recognition (AR) and provide context-aware assistance. Hence, the role of segmentation phase is to disentangle sensor events from smart environments into relevant sets of ongoing ADL to enable AR algorithm to analyse the data more efficiently as discussed in section 2.1.2.

Recent studies have mainly investigated in-direct methods to make decisions on which ongoing activities or new activity a given sensor observation belongs to, i.e., storing the data in the log/database first and then performing the queries to make segmentation decisions. For this, knowledge-driven (KD) and data-driven (DD) knowledge modelling approach discussed in section 2.2, time-series analysis using fixed/dynamic window size, and query-based techniques were commonly explored. However, little has been explored in developing direct methods to inspect and reason with individual sensor observations as they are registered by the data collection application. Also, the majority of the studies on data segmentation prescribe all users conduct ADLs with a generic set of actions which makes the system reusable with other users. Nevertheless, the diversity in the user's medical record, practices and cultural background influence the way a generic and personalised set of actions for ADLs are conducted. Hence, making segmentation approaches to discard user-specific actions and restrict the AR approach in the next step to detect and provide personalised assistance to the users.

Consequently, this chapter focuses on developing a segmentation approach which incorporates personalised user actions in knowledge modelling to the AR process. In addition, this chapter takes the motivation of state-of-the-art hybrid approaches discussed in section 2.2.3 forward, whereby, "*seed*" activity model is developed using KD approach which can later be evolved overtime using DD approaches in the future work. Therefore, this chapter focuses on recent studies adopting KD approaches to develop the seed activity model, which supports segmentation to disentangle sensor events to relevant activities as they unfold in smart environments.

The remainder of this chapter is organised as follows. The prevailing studies related to segmentation, semantical knowledge modelling and AR process are reviewed in section 3.2. As a result, a novel segmentation method and algorithm is then proposed in section 3.3. Next, the system implementation details and evaluation results are discussed in section 3.4 and 3.5. The chapter concludes and highlights future research directions in section 3.6.

3.2. Review of Semantic-based Data Segmentation Studies

Previous studies have applied approaches such as time series (fixed/dynamic time window [27], [140]), statistical and probabilistic [141] for data segmentation. However, they have failed to separate sensor observations based on the relationships described in the knowledge model to ongoing activities without storing the data first and then performing continuous queries. Therefore, KD approach has received an increasing amount of interest to express multi-layered relationships between sensors and domain-specific knowledge. The process of defining intricate sets of relationships between entities has been investigated in the past studies. These studies can be categorised as syntactical, semantical and pragmatic in information theory[142]. The characteristics of the syntactical approach are such that, a concept represented in a two or more non-syntactically equivalent statements are assumed to be statements or facts of independent concepts. In contrast, the semantical approach focuses on representing the meaning of a concept using relationships [142], [143]. Hence, the meaning of one concept can be syntactically represented in more than one statement with each referring to the same concept. The pragmatic approach, on the other hand, selectively studies the relations between a concept and inhabitant in a given context of interest.

The following sections review some of the recent studies tackling challenges of continuous data segmentation in relation to the semantical, syntactical and pragmatic information theories.

3.2.1. Semantical Approaches

Studies in [120], [144] adopted ontological models to describe ADLs, environmental entities and their relations to classify and infer unfolding activities. The classification approach performs the continuous queries on events stored in the database and knowledge model without using any automatic reasoners to determine the relationship between events and ADLs. Therefore, individual each sensor events are not directly inspected as they arrive and organise them in the new or ongoing set of activities. Likewise, effort in [145] proposed C-SPARQL, where individual sensor events in a stream are annotated with a timestamp and continuously queried using a specific window size. C-SPARQL query language is an extension to SPARQL Protocol and RDF Query Language (SPARQL) used for the graph-based database. The fundamental limitations of the approach are the classical multi-query optimisation problem. The common challenges in the multi-query problem are to identify the common parts, adapting/reformulating the order in which queries are executed with dynamically sliding window size.

Another branch of work, [24], [146], adopted Semantic Web Rule Language (SWRL) based inferencing rules to define the nature of activities with a temporal representation technique. These SWRL rules and Java Expert System Shell (JESS) rule engines were leveraged to segment the sensor events using their timestamp information and perform entailments for the complexity of the ongoing activities. One of the limiting factors of this approach is the use of generic ontology reasoner, which does not take full advantage of the expressive capabilities of the ontological model. Also, it is unclear if reclassification of the whole ontology is done incrementally or not. The incremental reclassification method available in the reasoner such as Pellet can help reason with the updated part of the ontology instead of the whole ontological model. Otherwise, non-incremental reasoning approach can degrade the performance and scalability exponentially with the increasing size of an ontological model. Another study, [69], adapted rules to generated to model generic and inhabitant specific ADL preferences. Yet, each time the new rules are added or updated to enrich the knowledge base, the whole ontological model is reclassified. Moreover, management of the models generated using the generic and inhabitant specific rules exclusively creates further complexity, specifically in the activity learning phase.

3.2.2. Syntactical Approaches

Work in [147] proposed a layered ontological model and complex event processing (CEP) engine based framework, namely, AALISABETH. This framework incorporates temporal reasoning with a dynamic time window size mechanism to segment the incoming data and perform AR in real-time. The framework leverages the Esper based CEP software and D2RQ engine to map data into RDF graphs. Even though the Esper CEP engine is highly optimised, scalable and open-source, the system falls short in directly segmenting the incoming sensor data semantically in real-time as it arrives from the sensor network. Thus, limiting the client applications to receive an event-based notification which is critical in an emergency scenario such as fall detection. Another significant limiting factor of the framework is events from the sensor network are continuously stored and queried from the traditional relational database management system (RDBMS), and the results are later mapped to the graph-based database. Instead of inspecting individual events directly, continuous queries are performed to obtain a set of sensor events between a specific time interval and then perform Web Ontology Language (OWL) based reasoning. Alternatively, the incremental Pellet reasoner can be further utilised instead of creating an overhead of continuous querying and mapping each sensor events from the RDBMS database using the D2RQ tool. Besides the overhead problems, the framework is also not developed with the consideration of user preferences when performing a generic ADL.

3.2.3. Pragmatic Approaches

Work in [148] presents an event filtering approach by adding preconditions with probabilities on the pre, during and post phases when carrying out each ADL to segment the incoming events. This approach has achieved good accuracy in segmenting and recognising mixed activities. Nevertheless, it is unclear how new activity is detected by the algorithm when an action is shared with more than one activity as an action can be part of a principal activity or precondition action for another activity. For instance, *MozzarellaCheese* can be part of the precondition of *MakePizza* ADL and postcondition for *MakeCheesyToast* ADL (assuming the cheese is left to melt on a pre-toasted bread). Alternative work in [149] leveraged evidential theory and proposed three segmentation algorithms based on location, activity model and dominant-centred (key actions for an activity) for non/-interleaved activities. The evaluation result indicated that location and activity model-based segmentation algorithms fall short in distinguishing activities when performed in the same location and with similar everyday objects for activities compared to the dominant algorithm. One of the critical limitations of all the three algorithms in [149] is the lack of support for user preferences and a reasoner to automatically detect and recognise the activity.

In general, the advantage of adopting the syntactical approach is that knowledge can be structured using defined syntax, queried and interpreted by the machine. However, the syntactical approach suffers from the flexibility of expressing intricacy of relationships and meaning between two concepts that pragmatic and semantic approaches can provide in a holistic view. The semantic theory is rooted from semiotics in philosophy which in broad is the study of signs and its significations (meaning)[150]. These signs can be based on words, images, sounds, gestures and objects. Henceforth, the semantical theory is studied heavily in cognitive philosophy, natural language and machine learning [151].

Based on the limitations identified above, this chapter makes five contributions by proposing. Firstly, a semantic-enabled segmentation approach is developed, which combines generic and personalised ADL knowledge that enables simple and mixed activities to be recognised in real-time. Secondly, a knowledge-based (KB) model is conceptualised by capturing the relationships between entities in the house and ADLs. Thirdly, a light-weight and pragmatic mechanism is proposed to manage and infer user specify ADL preferences. Fourthly, a semantical decision engine algorithm is developed to segment unfolding activities. Finally, system implementation details and evaluation results are provided for applicability of real-time data segmentation in the AAL system.

3.3. The Semantical Segmentation Approach

A semiotic theory inspired the segmentation approach is developed which examines the relationship of the sensor event with an everyday object and meaning of their actions from the known set of ADLs. Therefore, allowing actions performed in no particular order within a sequential or mixed activities scenario to be disentangled and separated. An ontology-based knowledge modelling process is presented in section 3.3.1, which captures the environmental context (i.e., ambient attributes, objects, location, and sensors network), generic and inhabitant specific preferences to perform ADLs. A segmentation decision engine is developed in section 3.3.2 which takes three inputs: a new sensor event, the ontological model and a set of previously segmented sensors for a given activity. A multithreading method is adapted to perform segmentation tasks of buffering sensor data stream, event recycling, decision engine, managing ADL threads and manipulating data from the Jena Fuseki [136] triplestore(TDB). This multithreading mechanism to segment sensor event based on semantics is described with a pseudo algorithm in section 3.3.3. Further information can be viewed in work [152], [153]

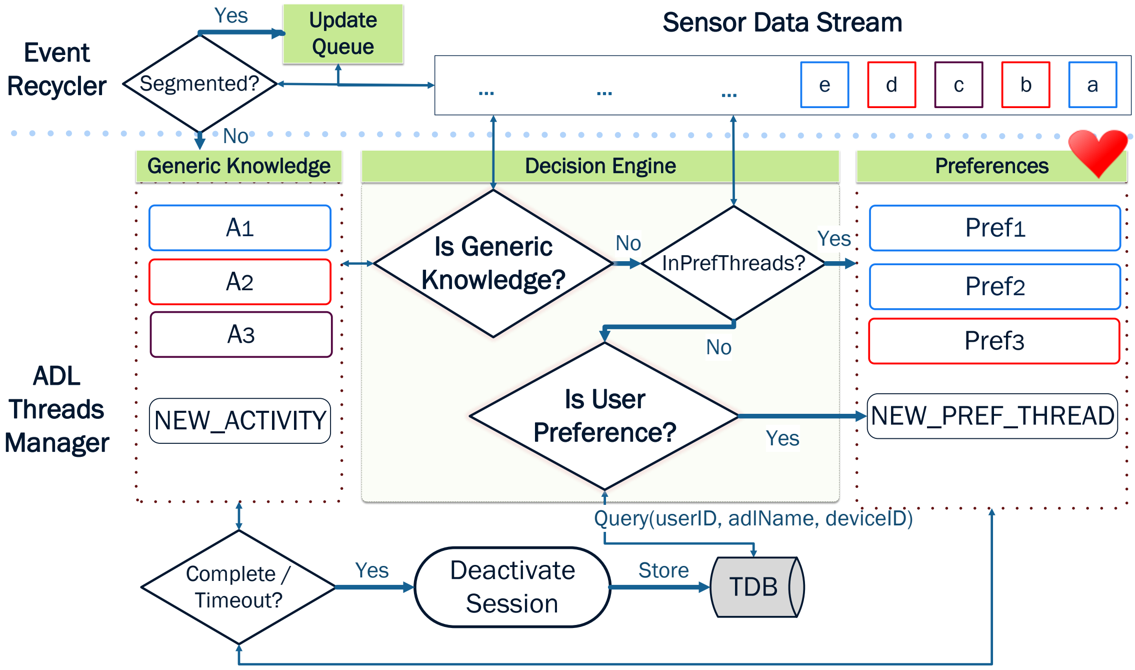


Figure 3.1. Overview of the semantically enhanced segmentation approach with generic knowledge (T-box) and user preferences (A-box) for reasoning.

Figure 3.1 demonstrates the overall semantical segmentation approach. On the top right of Figure 3.1, sensor events are initially added to the data stream. Both generic and preference ADL threads examine the sensor observations in the queue using decision engine and store the applicable events independently. Thus, allowing one sensor observation to be shared between multiple activity threads with dissimilar ADL goals. For example, opening the *Fridge* door action detected by sensor *e* at T_n can be shared with *MakeTea(*A_1*)* and *MakePasta(*A_2*)* ADL thread. The ADL threads manager generates a new ADL thread (*NEW_ACTIVITY*) if the sensor observation is not part of any ongoing ADL threads. The event recycler thread maintains the sensor data stream buffer and removes the processed sensor observation. There are two types of ADL threads being created to capture generic actions (sensor *b* attached to *PastaBag*) for a given activity (*MakePasta – *A_2), and user-preferred actions observations (sensor *d* attached to *HotSauce* at T_n) for that activity (i.e., Pref_1 - *PrefMakeVegPasta*). The decision engine infers if the new sensor event, along with the prior set of sensors for a given activity is part of the pre-defined generic set of actions by performing semantic reasoning and invoking queries to the TDB for user personalised actions for the activity of interest. The new preference thread (*NEW_PREF_THREAD*) is only created when the new sensor event is part of a personalised action for a given ongoing activity and if there is no active user preference thread. In CHAPTER 4 and CHAPTER 5, each continuous activity thread containing the set of segmented sensors data will enable validation of AR accuracy, timeout and completion procedures, i.e. storing and prompting the inhabitant.

3.3.1. ADL Knowledge Modelling

The fundamental building block of ADL modelling comprises of three phases; (1) environmental context (EC) modelling, (2) semantical relationships (SR_i) mapping and (3) bespoke user preferences ($Pref_j$) with objects and ADLs. In the initial phase, classes and instances are created to conceptually define the meta-/physical entities (ET_k) and their characteristics as classes (C) to form an overall environmental context (EC) for a given SH environment. The main entities described are: a person (X_n), rooms (Location, L_m), ambient characteristic (AC_p), sensor characteristics (S_o) and everyday non-/fixed objects (Obj_x); see equation 3-1.

$$EC = \{X_n, L_m, AC_p, S_o, Obj_x\} \quad \textbf{3-1}$$

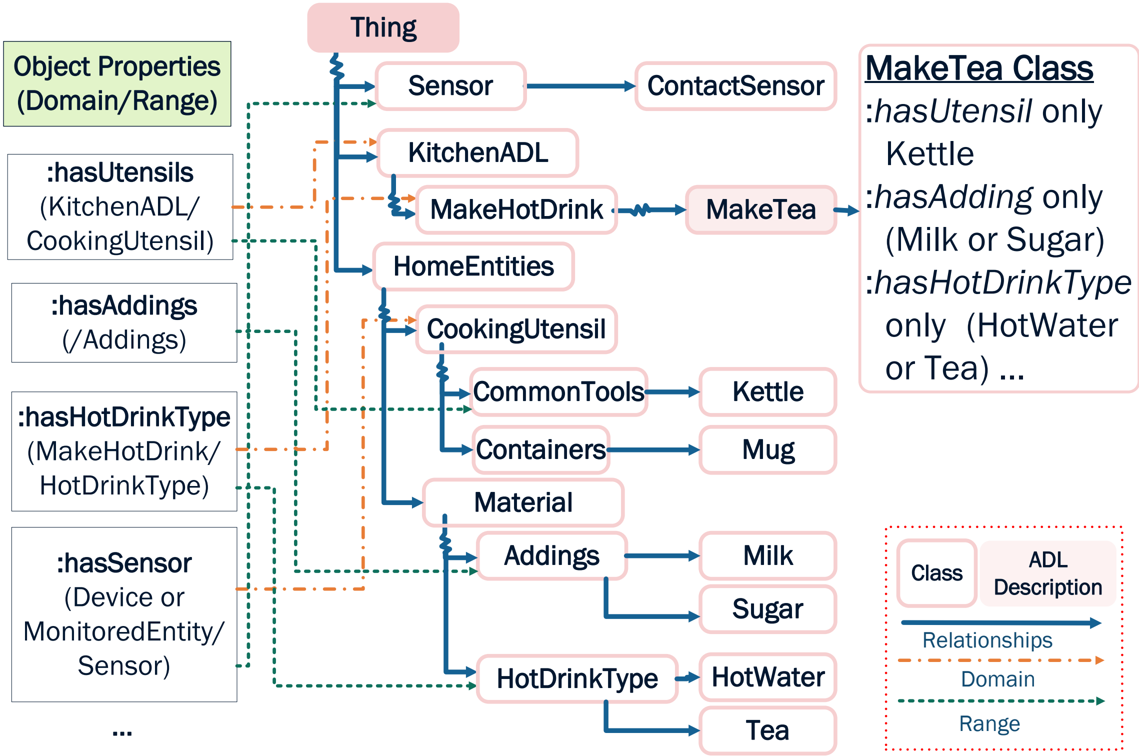


Figure 3.2. Semantical relationships between everyday objects, set of actions for *MakeTea* ADL and sensor characteristics.

In the second phase, the semantic relationship (SR) properties between EC classes (i.e., everyday objects) and ADLs are recorded. The instances of EC classes are then created for the sensor environment (SE) to add a relationship (R_e) between sensor, object it is attached to and object's use in ADLs; see equation 3-2 This abstraction in ADL's actions description allows decoupling, reuse and adding the further meaning of the actions to the activity using R_e. For instance, *MakeTeaADL* class is described using *hasHotDrinkType* (R) relationship property with Tea (C) and the characteristics of the $hasHotDrinkType$ property to be only used for *MakeHotDrinkADL (domain)* and everyday objects that are used for *HotDrinkType (range)*. Therefore, if no other ADL that is a subset of *MakeHotDrinkADL* that has a *hasHotDrinkType* property with *Tea*, it can be deduced that this action is potentially a part of *MakeTeaADL*. Similarly, other actions for *MakeTeaADL* can be described using *hasUtensil*, *hasContainer* and *hasAddings* properties for using the kettle and adding sugar and milk to the teacup. Figure 3.2 shows the relationships between a set of EC classes and *MakeTea* ADL to show the meaning of inhabitant's action.

$$SR = ADL_n(R_e, EC_n) \rightarrow R_e \rightarrow SE; \qquad \textbf{3-2}$$

Moreover, the sensor environment (SE) information is then encoded to describe an existing set of EC items available in the given residential environment and the sensor attached to it as instances (I_w). Therefore, instances of EC (iEC_w) such as everyday objects ($iObj_w$) and sensor (iS_o) with their relevant classes (C_n) are explicitly described with the relationship (R_e) between them initially. For example, *to₁* is an instance of *ContactSensor* (S) that *isAttachedTo* (R) a *RedKettleObj₁* ($iObj_w$) which is a class type of *Kettle* (Obj_x). The observed values/states of an iS_o are stored as primitive data types (pt_u) for a single observation or creating another instance of an observation class containing the primitive data for multiple observations; see equation 3-3.

$$SE = I_w(R_e, S_o) \longrightarrow R_e \longrightarrow I_w(R_e, ET_k) \,||\, I_w(R_e, \{pt_u\}) \qquad \textbf{3-3}$$

The final phase is to capture inhabitant specific preferences (*Pref*$_j$) that are subjective to the individual's cultural background, and rituals followed in carrying out a given ADL. It is essential to keep the generic (factual and commonly accepted by the wider community) and personalised sets of ADL description disjointed to avoid generalising or assuming both must be actioned to complete the activity. Therefore, instances that are members (R_e) of *Preference* and ADL_n classes are created to capture actions or ambient attributes using iEC_w that are specific to a person (X_s); see equations 3-4 and 3-5. For example, an individual *Bob* (I) who is a type of $Male$ (C) has a set of instances of *Preferences* that are linked with *hasPreference* relationship (R). An example of a preference instance is *BobMakeSpicyTeaPref* ($Pref$) which is a type of *Preference* (C) and *MakeTeaADL* (ADL) with a set of iEC instances, i.e., GingerObj(I) and *CinnamonObj*(I). This statement means that {*Bob*} has a preference to make tea and he may/like to put ginger and cinnamon in his tea.

$$X_s = I_w\,(R_e, Human \sqsubseteq Male) \rightarrow R_e \rightarrow \left\{Pref_1, \ldots Pref_j\right\} \qquad \textbf{3-4}$$

$$Pref_j = I_w\,(R_e, ADL_n \cap Preference) \rightarrow R_e \rightarrow I_w\,(R_e, iEC_w) \qquad \textbf{3-5}$$

3.3.2. Semantic Decision Engine

Central to the semantic-based decision engine is the ability to identify relationships between the sensor, everyday object and actions described in ADLs based on the ontological model and triplestore querying. Hence, allowing the decision engine to segment user actions performed in any order for single or multiple ADLs in a mixed activities scenario. The generic ADL actions are automatically recognised using terminology box (T-box) reasoning method with incremental Pellet reasoner and inhabitant specific actions using assertion box (A-box) reasoning method. The decision engine is utilised by individual activity threads to find an association with new, previously observed events and candidate ADL class. The classification of candidate ADL class is continuously updated and refined with further evidence of actions that satisfies the ADL descriptions.

The decision engine takes three inputs, processes them into two stages and outputs the updated results. The three inputs are (1) semantic-based KB model created in section 3.3.1, (2) activity thread (AT_n) attempting to find relations with the (3) new sensor event (e_m). Each AT_n contains structured information about generic and preferred actions observed as sensor events, ADL class and list of preferences matched that are associated to the inhabitant. The two-stage decision-making process updates the activity thread accordingly as the new sensor events are inspected incrementally for any association.

$$\begin{aligned} AT_n = \{ & tbox[class{:}\, someADL, s\{\ldots,\, e_m\}\}], \\ & abox[Pref_j[name{:}\, somePref, s\{\ldots,\, e_m\}]]\} \end{aligned} \quad \textbf{3-6}$$

In the first stage of the decision-making process, generic semantical relationships are traced from EC to SR and SR to SE compared to inverse when developing the KB model [154]. Therefore, the metadata of a sensor observation e_m is analysed to find the ET the sensor is attached to and deduce the potential R_n with a set of ADL_n description. This metadata within KB consists of relationship properties such as domain and range for a given ET. Therefore, the association between ET_k (i.e., everyday objects) and ADLs can be automatically inferred using semantic reasoners or simply querying the KB model. This process is known as terminology box (T-box) reasoning [155].

The second stage is only executed when the result returned from T-box reasoning identifies any conflicts with the ADL class description. The conflicts can be raised when a given sensor attached to an ET is forced to be part of a given ADL which is outside the restricted set of ET_k. In this case, it is assumed that ET is part of inhabitant's preferences or part of a new set of actions for ADL_n. The preferences are currently pre-defined and stored as individuals containing a list of iEC_s that an inhabitant prefers to use to perform a given ADL. Therefore, semantic queries are made to extract all preferences of the inhabitant (*userID*) for a given ADL (*adlName*) that as sensor observation (*deviceID*) as an action. This process is known as assertion box (A-box) reasoning.

The semantic reasoner carries out several tasks using T-box and A-box knowledge which includes but not limited to: satisfiability, subsumption, consistency checking equivalence, disjointness, and instance checking [154], [156]. The satisfiability task is to ensure the class description (axioms) is not contradictory. The subsumption task ensures class B satisfies all the inheriting properties (R) of parent class A. The consistency checking reviews axioms descriptions for classes and their instances for any violations in the class definition. The instance checking ensures the relationships with other instances are within the boundary of a set of classes it can subsume. The equivalence task is to match the two concepts concerning its properties in contrary to disjointness tasks. The conjunctive querying answering is performed at the second phase of decision engine to identify inhabitant's preferences with a given ET using relationships between instances of EC and ADLs.

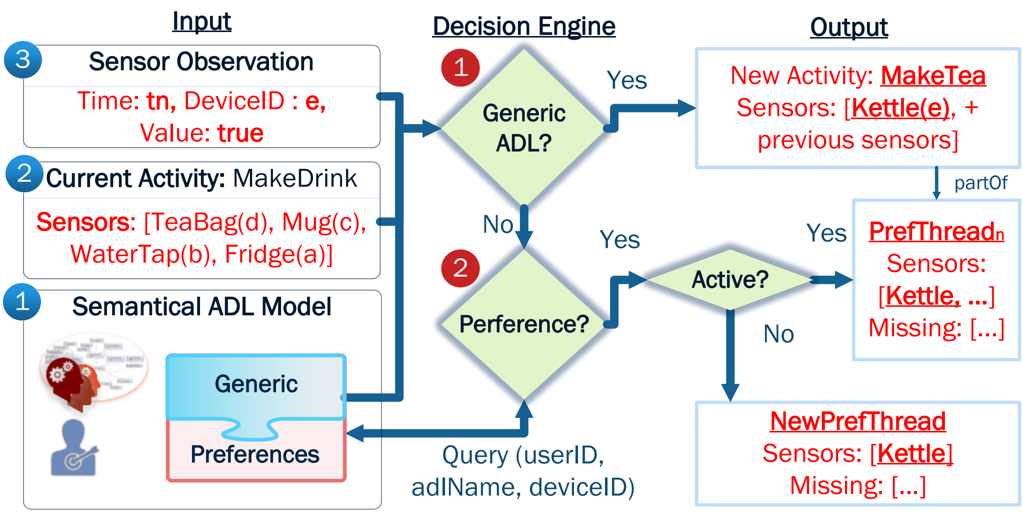


Figure 3.3. Semantic-based Decision Engine; Input: new sensor observation (e_5), current activity with a set of sensors and semantical ADL model, Output: new activity result.

Figure 3.3 illustrates the three inputs taken by the decision engine to verify if the new sensor observation Ginger(e_5) is part of the generic/personalised action of the ongoing *MakeTea* activity (AT_1). Initially, a new activity thread, AT_1, is created to add the first sensor observation, *Fridge* (e_1), into the empty set of sensors and the results returned from the two-stage reasoning process. In this case, e_1 is inferred by the generic T-box reasoner to be part of *KitchenADL* in the first stage of the decision engine. As the new sensor event, e_2 occurs, the current AT_1, temporarily add it to the list e_1, e_2 and perform the generic reasoning again with the same activity result. Therefore, the action is part of AT_1 but more than one sub-activities share the same actions. Similarly, other events are added to $AT_1 = \{e_1, e_2, e_3, e_4\}$ as they occurred with new *MakeTea* activity name which is a descendant class of *MakeDrink* and *KitchenADL*. Until now, only the first stage of the decision process is performed due to the generic nature of the ADL actions. The next sensor observation, e_5, is attached to *Ginger* running any personalised actions. The activity name, *MakeTea* of A_1 and the new sensor observation Ginger(e_5) is used to perform subsumption reasoning in the first stage of decision engine and returned inconsistency in ADL description error. In the second phase, the decision engine checks if the *Ginger*(e_5) sensor is part of an inhabitant's preference(s) stored in the triplestore and add it to A_1. In this case, *spicyTea* preference was identified and as there were no sub-activity preference threads already active for A_1, new thread $Pref_1$ was created along with other missing *spicyTea* actions.

$$AT_1 = \{tbox\,\{name{:}\,makeTea, s{:}\,\{\,e_1, e_2, e_3, e_4\},\ \&\ abox[Pref_1[name{:}\,spicyTea, s{:}\,\{e_5\}, missing{:}\,\{\ldots\}]]\,\}. \quad \textbf{3-7}$$

3.3.3. Segmentation Algorithm

The algorithm in Table 3.1 shows the segmentation process, use of decision engine (DE) and multithreading mechanism discussed in section 3.3 to separate sensor observations. The ADL threads manager class performs the algorithm, and it is broken down into three stages. The first stage is to iterate over all the active T-box threads (AT_n) and use the current list of sensor observations in each thread along with the observed sensor event (e_m) being investigated to refine a ADL inferencing result or assume sthe tart of new ADL. For simplicity, the pseudocode shows only the first iteration AT_1 is conducted. Line 1 checks if there is $\neg\exists e_m$ in AT_1 then perform T-box and A-box reasoning in stage two and three. Otherwise, e_m is assumed to be the start of new ADL activity. Hence, new AT_{n+1} is created with e_m in line 12. The T-box reasoning task in line 2 is performed by calling DE by taking three inputs: e_m, set of current sensor events in AT_1 and $T = \{EC, SR, ET\}$ in KB. The new deduced ADL result (*Class* c) is evaluated for conflicts and if $c \sqsubseteq \mathrm{current}AT_1$ class then AT_1 is updated with c along with e_m; see lines 3 and 9.

In the second stage, inhabitant's preferences are checked when conflicts in the result are detected. All the A-box threads are checked if e_m is part of active preference thread then add the event to $AboxT_a$ thread. Otherwise, any inhabitant (*userID*) preferences ($AboxT_a$) of a given ADL class c inferred for AT_1 is queried from the TDB, and new A-box threads are created if matched; see lines 4-7. The final stage is where all the housekeeping for the sub-threads and the process of re-evaluating the session timeout window size and timeout cases based on the data of the segmented set of observations. Details of the semantical segmentation mechanism can be found in our previous work [152], [153].

Table 3.1. Pseudocode for Semantical Segmentation Algorithm

Input: $e_m, T = \{EC, SR, ET, AT\}$, *name*, *userID*
Output: void

if $\neg\exists e_m : AT_1$ **then**
    Class c = DE.runTbox(e_m, AT, T) */* 1) T-box reasoning */*
    if $\neg\exists c \sqsupseteq AT_1$ **then**
        if $\neg\exists AT_1.AboxT_a(e_m)$ **then**
            $AT_1.AboxT_a.add(e_m)$ */* 2) A-box reasoning */*
        else if $\neg\exists DE.queryTDB(e_m, AT_1, name, userID)$
            $AT_1.addAboxT(e_m)$ */* 2.1) create A-box thread */*
    else
        $AT_1 \equiv c(e_m)$ */* 1.1) update ADL classification */*
    end
else
    $AT_{n+1}(e_m)$ */* 1.2) create new T-box thread */*
end
closure (AT_1) */* 3) activity completion and timeout procedures */*

3.4. System Implementation

An android mobile application and RESTful web service have been used to create a service-oriented architecture (SOA) system. An SOA enables the web service to execute computation tasks such as segmentation and AR on the sensor events stream and store the results into the Jena Fuseki triplestore [136] using Jena API. The web service exposes these resources to multiple client devices running on independent operating systems using hypertext transfer protocol (HTTP) asynchronously. The web service receives all the sensor events from the sensing environment using wired/wireless connections methods and performs four primary tasks; broadcast, store, segment sensor events and performs AR. The sensing environment is capable of collecting ambient data using off-the-shelf binary and multi-sensors supported by Securifi Almond router with ZigBee, Z-wave and Wi-Fi communication protocol. Also, dense sensing is supported by miniature IoT boards that are based on Arduino microcontroller with radio frequency (RF) and Wi-Fi capabilities to transmit and collect analogue/digital sensor data; more details in [157]. The sensor observations and the results from segmentation and AR are broadcasted independently using server-sent (SSE) protocol and stored in the Apache Jena TDB and exposed using the Fuseki server. Multithreading concepts have been employed to segment each ADL into a thread described in section 3.4.2. A single ADL thread runs the T-Box reasoning and one or more A-Box thread(s). The reasoning result and sensor events are broadcasted to the clients and the Android application continuously capture and presents the information to the inhabitant. Figure 3.4 shows a snapshot of how concurrent actions of three activities are separated into different threads and presented on the Android application. Details of the SOA implementation and multithreading concept can be found in previous studies [157], [158].

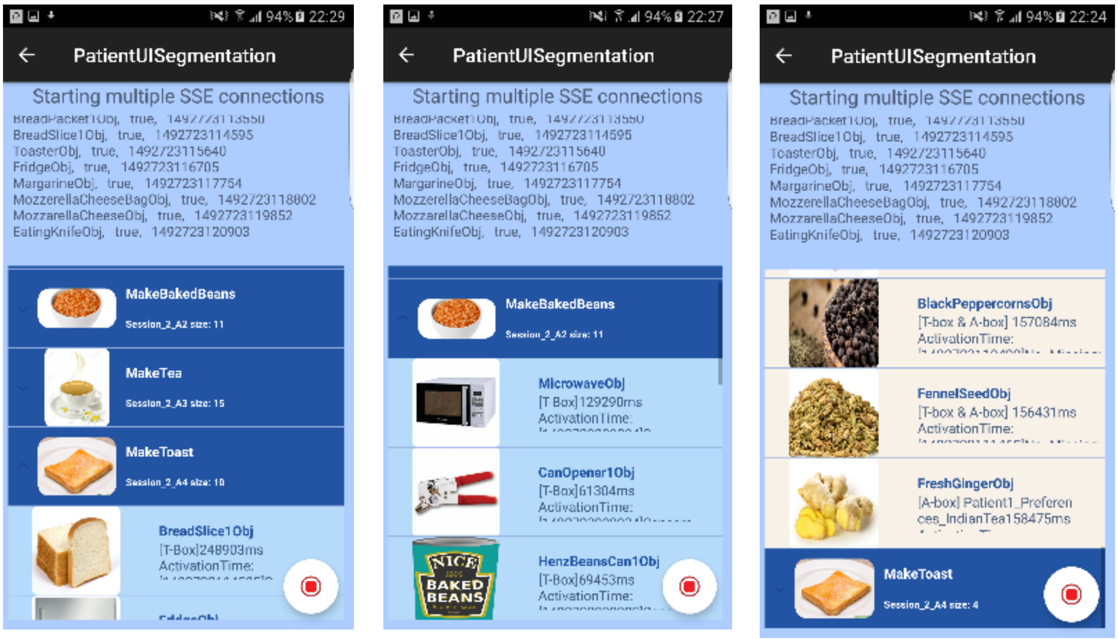
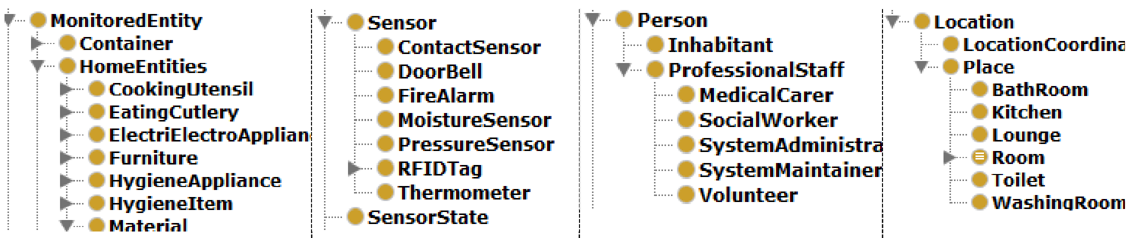


Figure 3.4. Segmentation results for three concurrent ADLs

3.4.1. Ontological Modelling

The generic knowledge for segmentation is represented using the semantic web framework. This framework provides web ontology language OWL to formally express the complex knowledge into classes, relationships (object & data properties) and data (individuals) [116]. In addition, standard vocabularies are used to represent the KB and encourage sharing across applications to create an ever-growing, human and machine-readable web of knowledge. There are many automatic reasoning tools available to read this KB to identify inexplicit facts based on relationship definition and the section 3.4.3 elaborates on the selection of a reasoner. The main goal of the ontological model is to express what, where and how the actions for ADLs should be fulfilled. For this, *EC*, *SR*, and *Pref* are modelled in three phases using ontology editor tool named Protégé[159]. Initially, *EC* concepts such as everyday objects, person, sensor characteristics and location were modelled as classes. Figure 3.5 illustrates the fragments of *EC* classes and their subclasses.



a) Everyday objects (Obj_x) *b) Sensor types (S_o)* *c) Person (X_n)* *d) Location (L_m)*

Figure 3.5. Conceptualising environmental context (EC) into Classes

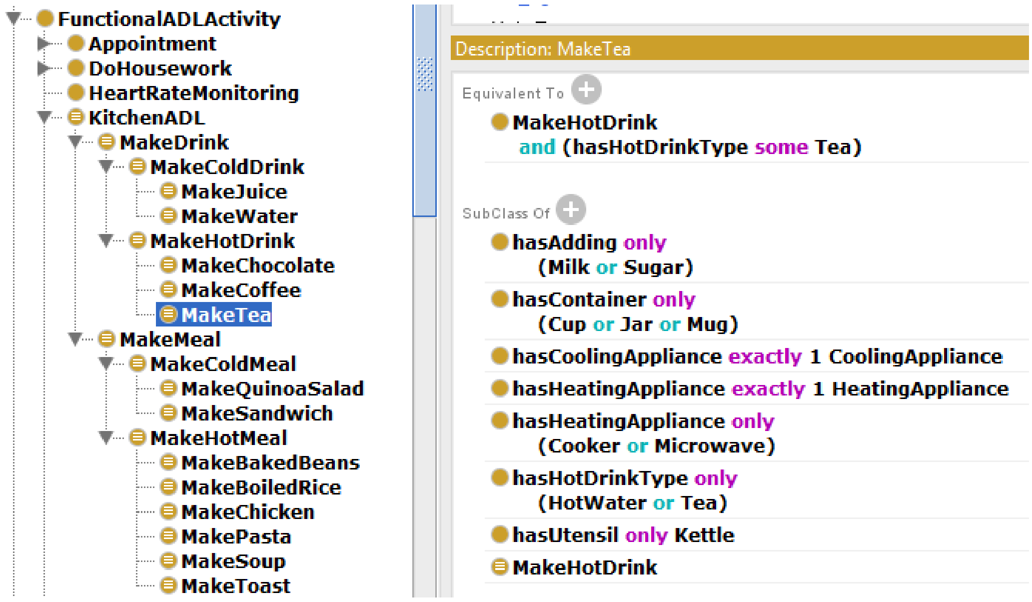


Figure 3.6. Partial description of *MakeTea* ADL with Semantic Relationship (*SR*) with environmental context (*EC*) in Protégé.

In the second phase, the *EC* classes are used to define *SR* between ADL classes and describe their actions iteratively using object properties. Figure 3.6 partially represents the *MakeTea* ADL in Protégé. The *MakeTea* ADL class inherit the properties described from super-classes and uses *rdfs:subClassOf* object property to define actions or the context to carry out the activity. The actions properties and the classes of everyday objects for the *MakeTea* ADL are described using object properties *hasAdding*, *hasContainer*, *hasHeatingAppliances*, *hasHotMealMaterial*, and so on. These object properties can have characteristics and relationships between everyday objects classes and the ADLs. For instance, *hasHotDrinkType* object property has a *domain* of *MakeHotDrink* ADL class and *HotDrinkType* material as *range* property. Therefore, if any everyday object that is a subclass of *HotDrinkType*, then that object is part of the actions defined for *MakeHotDrink* ADL class or its subclasses. These object properties are used to add further restrictions such as universal and existential quantification (∀, ∃) using some and only, logical operations such as not, and, or (¬, ∧, ∨), and cardinality restrictions (≤, ≥, =). Other standard operators are also available and can be used to increase the expressivity of the ADL model in terms of class, relationships and data. Similarly, the other 12 subclasses of *MakeDrink* and *MakeMeal* ADL classes are also described with relevant relationships. As multiple relationships with ADLs and everyday objects are created, the observed data (defined as individuals) with a set of assertation statements containing everyday object and object properties are used by the reasoning engine to automatically infer the type of the ADL class the actions in the individual belongs to.

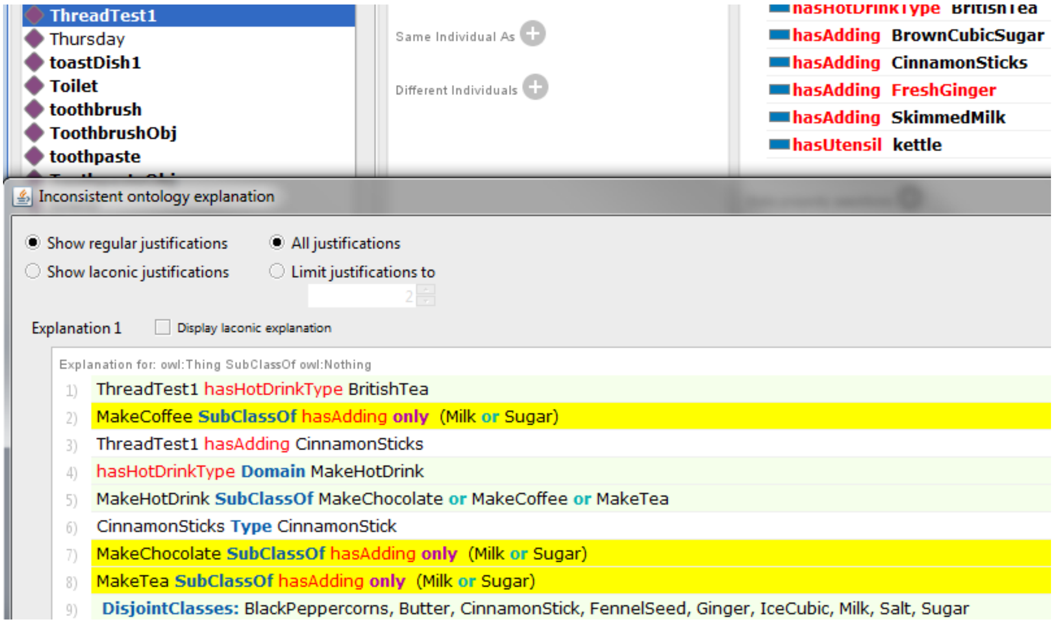


Figure 3.7. Inconsistency on *hasAdding* object property due to the restriction applied to *MakeTea* ADL class.

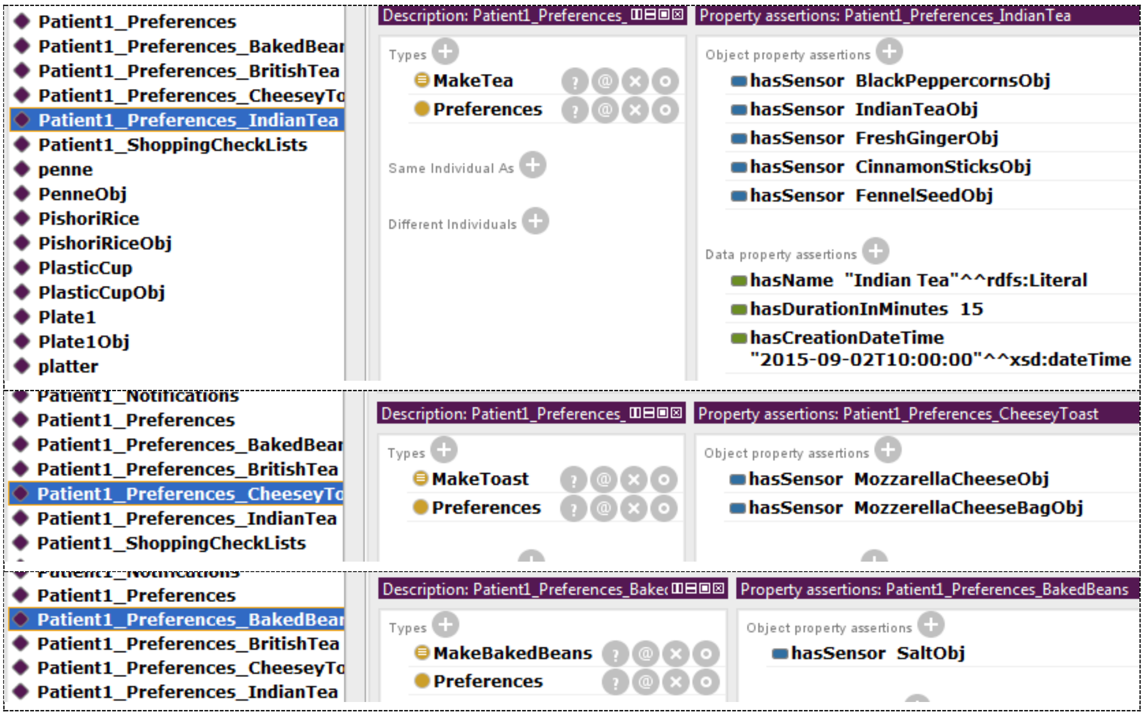


Figure 3.8. Inhabitant preferences as individuals with a list of sensors.

Finally, the inhabitant specific preferences (A-Box) are captured by creating individuals with a direct relationship with instances of sensors to avoid the inconsistency in ontology description for generic knowledge. In the generic knowledge, not all adding (ingredient) for *MakeTea* ADL are defined, and ingredients such as *FreshGinger* and *CinnamonSticks* are subjective to the individual. Hence, forcefully adding ingredients in an instance that is the type of *MakeTea* ADL will result in the inconsistent ontology as highlighted by the explanation window in Figure 3.7. Therefore, instances of preferences are associated with the inhabitant and to a given ADL class which has a list of sensors that are attached to the everyday objects and other attributes. Figure 3.8 presents an example of three inhabitant preferences. The top section presents individual named, *Patient1_Preferences_IndianTea*, which has a type of *Preference* class for *MakeTea* ADL class along with a list of sensors using *hasSensor* object properties and data properties to describe other attributes such as preference name and creation timestamp. Similarly, different preferences are shown in the middle and bottom of the figure to specify *MakeToast* and *MakeBakedBeans* preferences.

Another method is available to layer the inhabitant specific and generic ADL ontology descriptions along with SWRL rules. For this, OWL API and Jena API can be used to create and manipulate the model once generic, and inhabitant specific models are combined, and rules are loaded into the memory. The reasoning can be performed using the Pellet reasoner and JESS rule engine after combining the generic and inhabitant specific ontology that is managed dynamically. However, the main limitation of this method is that the changes made to the inhabitant specific ontologies will need to be tracked along with the mechanism to resolve any conflicts in the knowledge that may arise. Additionally, inhabitant specific reasoner will need to be created and maintained [160] at run-time. Hence, the amount of in-memory space, the number of processing cores and computation power required can grow exponentially. As a result, it can create high latency in segmenting individual sensor events and undermine the scalability of the approach. Therefore, the first method is selected as it is lightweight, and no inhabitant specific reasoner is required to be running. The SPARQL Inferencing Notation (SPIN) [161] rules or just a SPARQL query language can be executed on the triplestore to retrieve multiple inhabitant's preferences for a given ADL class simultaneously. Therefore, this method is considered appropriate during the segmentation phase as the inhabitant's preferences can be scalable and has lower latency in terms of query time, and there are no additional overheads for running multiple reasoners per inhabitant.

3.4.2. Multithread Segmentation Process

The multithreaded segmentation processes are depicted in Figure 3.9, where actions for *MakeTea* and *MakeToast* ADLs are performed concurrently. The generic and preferred actions are observed at a given time (t_n). The T-box activity thread (AT_1) is initially created when the *cupObj* sensor is activated at t_1. The AT_1 continuously stores the events into the thread if the decision engine infers an association with generic ADL class in the ontological model or personalised preference(s). The object attached to the *cupObj* sensor is queried from the triplestore, added to new individual and incremental T-box reasoning is conducted. The T-box reasoning result indicates that the object is related to *ADLActivity* class with no conflicts with the model, hence the A-box reasoning is not required to be executed. Next, the sensor event at t_2 is received, and AT_1 performs T-box reasoning with observed sensor *fridgeObj* along with previous sensor(s), in this case, *cupObj*. The decision engine returned a new result, *KitchenADL* class and it was compared against the current *ADLActivity* class for equivalent or subsuming class. In this case, the subsuming condition is satisfied and stores the *cupObj* and *fridgeObj* sensor events in the AT_1.

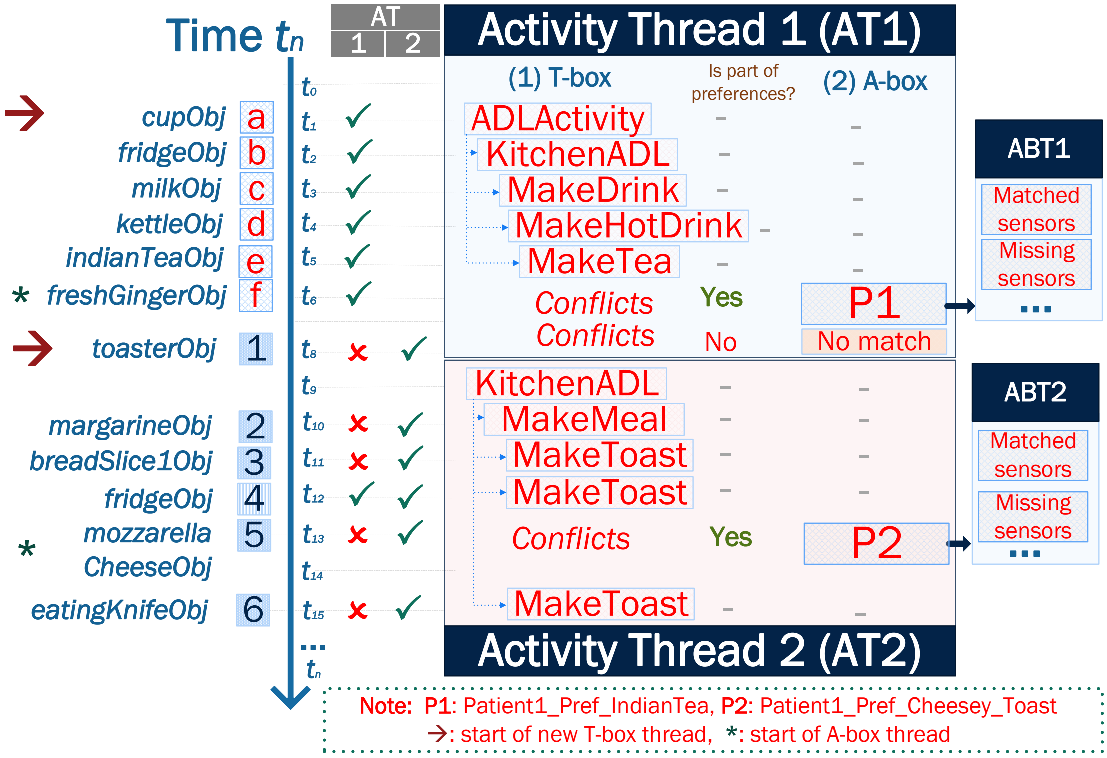


Figure 3.9. Concurrent actions for *MakeTea* and *MakeToast* ADL and segmentation process to create generic (AT_1 and AT_2) and preference (APT_1 & APT_2) threads when required

Similarly, *milkObj*, *kettleObj* and *indianTeaObj* sensor events are processed by AT_1 where the ADL classes are incrementally classified, and the sensor events are stored in the thread. Since the *freshGingerObj* sensor event is not described as part of a set of adding in the generic *MakeTea* ADL description, the decision engine returns with traceable conflicts. The decision engine then performs A-box reasoning to find any inhabitant's preferences related to *MakeTea* ADL containing *freshGingerObj*. Multiple preferences could be returned, in this case, only one preference named, *Patient1_Pref_IndianTea* (P_1) is returned as a result of SPARQL query. A single A-box sub-thread (APT_1) is created with other missing sensors and other relevant information from the preference into the thread. The APT1 thread then inspects the incoming sensor events and updates the missing and matched sensors list independently. AT_1 thread and the sub-thread(s) for A-box reasoning can continue examining unfolding events in the data stream until the completion criteria are satisfied, i.e. having no child ADL class and missing sensors in A-box threads or a dynamic timeout mechanism for the ADL. The completion/timeout criteria for the ADL will be inspected in future work.

The next set of actions for *MakeToast* ADL are observed between t_8-t_{14} and inspected by AT_1, but only one shared *fridgeObj* event is stored. The ADL manager running in parallel examines the sensor events in the queue and detects *toastObj* is not part of the *MakeTea* ADL class in AT_1 and APT_1 threads. Therefore, another T-box activity thread (AT_2) is created *MakeToast* ADL as depicted at the bottom-right of Figure 3.9. The same process is described for AT1 is executed for the AT_2 thread to capture events from t_{10}-t_{15} to AT_2 thread with one conflicting *mozzarellaCheeseObj*} observation. Therefore, the APT_2 thread is created when identified by the decision engine that *mozzarellaCheeseObj* is part of *Patient1_Pref_CheeseyToast* (P_2) to perform the *MakeToast* activity.

3.4.3. Reasoner and Supporting Tools

A reasoner is a software tool developed to perform A-box and T-box reasoning by the decision engine to perform tasks such as a consistency check of the ontological model and derive new facts from the KB dataset. There are several reasoners developed over the years, and most of them support first-order predicate logic [154] reasoning or procedural reasoning (perform forward and backward chaining). Some of the essential requirements for selecting a reasoner are that it supports the incremental classification for only the part of ontology that was affected by the changes [162], full description logics (DLs) family support for higher expressivity, rules support, justification of conflicts, low latency in classification and support both T-Box and A-Box reasoning. Studies in [154], [155] describe many popular reasoners using large ontologies, compare against their key features and categorise according to their characteristics. The incremental Pellet reasoner has been selected as it supports most requirements stated above, along with being open source and supported by multiple application programming interfaces (APIs) and ontology editors such as Protege and NeOn toolkit. OWL API and Jena API both support the Pellet reasoner to perform reasoning programmatically, querying and KB manipulation. Jena API further supports other reasoners to be Integrated easily. Although, the pellet reasoner takes up higher heap space and has higher delay time than FaCT+ when performing concept satisfiability checking after classification but outperforms in subsumption query [154].

3.5. Evaluation

3.5.1. Experiment Design

The actions for three ADLs are scripted in no particular order to perform with only generic actions and another with the inhabitant's preferences; namely, *MakeTea*, *MakeToast* and

MakeBakedBeans. The relevant actions for the generic(G) ADL and some inhabitant's preferences (P) are described in Table 3.2. These three ADLs are first tested individually in random order and then combined to create mixed activities scenario; incremental, concurrent and parallel; see Table 3.3.

Table 3.2. Single Activity Sequences Example

Activity	Type	Related actions/ sensors attached to objects	#
Make Tea	G	KettleObj, Cup1Obj, TeaJarObj, IndianTeaObj, KitchenSinkTap1Obj, SugarJarObj, FridgeObj, Milk1Obj, Spoon2Obj	9
	P	[FreshGingerObj], [CinnamonSticksObj], [BlackPeppercornsObj], [FennelSeedObj]	4
Make Baked Beans	G	Spoon1Obj, HenzBeansCan1Obj, HenzBeansObj, CanOpener1Obj, MicrowaveBowl1Obj, MicrowaveObj, Plate1Obj, EatingKnifeObj	8
	P	[SaltObj]	1
Make Toast	G	Plate1Obj, BreadPacket1Obj, BreadSlice1Obj, ToasterObj, FridgeObj, MargarineObj, EatingKnifeObj	7
	P	[MozzerellaCheeseBagObj], [MozzerellaCheeseObj]	2

Note: Generic (G) / Preference (P) actions, [SensorName] - User preference item, # - number of sensors

Table 3.3. Combinations of Simple Activities

Activity Comb.	ADL Sequences	Expected threads	no. Actions Gen. (G)	+ pref. (G+P)
AC1	MakeTea, MakeToast	2	16	22
AC2	MakeTea, MakeBakedBeans	2	17	22
AC3	MakeToast, MakeBakedBeans	2	15	18
AC4	MakeToast, MakeBakedBeans, MakeTea	3	24	31
AC5	MakeBakedBeans, MakeTea, MakeToast	3	24	31
AC6	MakeTea, MakeToast, MakeBakedBeans	3	24	31
Total		15	120	155

A total of 30 activity scenarios (6 for single and 24 for mixed activities scenarios for both G, and G+P actions) were created for the experiment and a thread simulated each scenario with sensor events occurring at 10ms interval. The sensor events contained a timestamp, name, sensor type, and binary data. The degree of accuracy to recognise an activity scenario is calculated in percentage by matching and tallying actual sensors events segmented correctly, and it divided by the total number of sensors events activated for each ADL. The average classification time is calculated by taking sensor observation segmented time by the reasoner minus the sensor observation time recorded for each activity scenario. The unexpected sensor observations within the activity scenario are omitted and recorded separately when calculating the accuracy and average classification time for the activity. Equally, several duplicate activity threads created in the activity scenario are also registered to see the effect on the overall classification times. The Samsung S6 edge smartphone running 6.0.1 Android OS was used, and the web service was deployed on the HP EliteBook Folio 1040 G2 with the i7 2.60GHz processor, 2 cores, 4 logical processors and 8GB RAM. The binary sensor events are currently simulated due to a limited number of sensors and time.

Table 3.4. Single Activity performed in no specific order with generic and personal preferences

Activity	Type	In relevant thread	Unexp. actions in thread(s)*	Excess thread (s)	Avg. time (ms) +
MakeTea	G	9	0	0	2394.67
MakeToast	G	7	0	0	2468.57
MakeBaked Beans	G	8	0	0	2372.25
MakeTea	G+P	13	0	1	10828.85
MakeToast	G+P	9	0	0	3786.87
MakeBaked Beans	G+P	9	0	0	1972.44
Total	**6**	**55/55**	**0**	**1**	**3970.61 (avg.)**

Note: * excludes additional thread(s) actions, + including excess threads

Table 3.5. Multiple activities performed in a mixed activities scenario

	Activity Comb.	Type	All actions in the thread(s)?	Excess thread(s)	Unexp. actions in the thread(s)*	Total Avg. time+ (ms)
Inc.	AC1	G	✓ 16	1	1	36330.64
	AC2	G	✓ 17	1	4	41543.17
	AC3	G	✓ 15	1	1	30354.98
	AC4	G	× 15/24	3	3	95819.25
	AC5	G	✓ 24	1	5	60742.14
	AC6	G	✓ 24	1	6	72690.97
	AC1	G+P	✓ 22	1	1	54949.21
	AC2	G+P	✓ 22	0	5	21905.05
	AC3	G+P	✓ 18	0	1	12561.28
	AC4	G+P	× 31	3	3	99807.19
	AC5	G+P	× 30/31	1	4	62016.20
	AC6	G+P	✓ 31	1	3	87298.32
Con.	AC1	G+P	✓ 22	1	0	56752.83
	AC2	G+P	✓ 22	1	5	23993.51
	AC3	G+P	✓ 18	2	1	64074.61
	AC4	G+P	✓ 31	1	1	70289.79
	AC5	G+P	✓ 31	2	6	131784.92
	AC6	G+P	✓ 31	2	5	181894.97
Par.	AC1	G+P	× 21/22	2	0	43055.55
	AC2	G+P	✓ 22	0	3	8309.10
	AC3	G+P	× 16/18	1	0	35944.94
	AC4	G+P	✓ 31	1	4	63737.04
	AC5	G+P	✓ 31	1	5	77355.87
	AC6	G+P	✓ 31	1	4	59173.90
	Total	**24**	**572/585**	**29**	**71**	**62182.73(avg.)**

Note: * excludes additional thread(s) actions, + including excess threads

3.5.2. Results

The average segmentation time taken per sensor event for single activity is 3971ms in contrast to 62183ms for mixed activities scenarios, as shown in Table 3.4 and Table 3.5. The result in Table 3.4 shows that all the sensor events for a single activity case scenario were adequately placed in the correct thread with 100% accuracy. Only the *MakeTea* activity case scenario created more threads with double the average time when processing 9 generic actions and 4 preferred actions. On the other hand, Table 3.5 shows 20 out of 24 activities performed in a mixed activities scenario or 572 out of 585 sensor events were added to the relevant thread, giving 97.8% accuracy. However, the segmented activity threads captured a total of 71 additional unexpected sensor events in the segmented threads which are not necessarily incorrect, i.e., multiple spoon objects or heating/cooling appliances when performing multiple activities interweavingly. Furthermore, 29 extra threads were created and failed to classify any ongoing activity.

3.5.3. Discussion

Although previous studies use varying ADL models, datasets, sensors and platforms, use scenarios, etc., the key features and outcomes for the recent KD studies presented in section 3.2 are discussed instead. The accuracy of single and mixed activities segmentation for evidential theory-based approach [149] is 81.8% and 76.2% on average and ontology and temporal [24] achieved 100% and 88.3%, respectively. Therefore, there is significant evidence that the proposed approach improves the accuracy of sensor segmentation with 100% and 97.8%, respectively. Also, user-preferences are taken into consideration by adopting the basic query-based approach and automatic Pellet reasoner for generic KB reasoning compared to their counterparts which adapt solely query-based approach inheriting classical multi-query optimisation problem in [145] and [147]. Nevertheless, one of the benefits for adapting multi-query approach is that higher performance and scalability can be achieved, however, suffer from the expressivity capabilities of KB due to explicit query development/maintenance efforts and the ability to use automatic reasoners.

The proposed method in this chapter seeks to strike a balance between automation by taking advantage of expensive ontology with incremental Pellet reasoning feature and performance of a query-based approach to managing the changing user-preferences. The average segmentation time information is not available in the presented KB studies; however, the proposed approaches observes 3971ms and 62183ms with sensors events activated at the 10s interval for simple and mixed activities scenarios. These results are still not suitable for the real-time system at this stage. However, the optimisation opportunities such as multi-thread safe reasoning [163], ADL threads management, parallel programming, partitioning workload to graphics processing units (GPUs) [164], and using a machine with a higher number of cores (i.e., quad-core, octa-core CPU or higher) to support more concurrent or parallel threads execution at the same time remain an open challenge. Table 3.6 presents a summary of the critical components of the recent KB studies presented in section 3.2 against the proposed semantical segmentation approach in this chapter.

Table 3.6. Summary of recent KB approaches

Studies (by year) / Features	**C-SPARQL** [145]**, 2010**	**Evidential theory** [149]**, 2013**	**Onto. and temporal** [24]**, 2014**	**AALIS ABETH** [147]**, 2015**	**Proposed**
Knowledge expressivity	High	High	High	High	High
SPARQL query support	Yes	Yes	Yes	Yes	Yes
Automatic reasoner support	No	No	Yes	No	Yes – incr. Pellet
Direct stream inspection	No	Yes	Yes	No	Yes
RDF stored	Yes	NA	Yes	No	Yes
User prefs. support	No	No	No	No	Yes
Sliding window support	Yes - Fixed-size	No	Yes	Yes	No – future work
Potential scalability issue	Low	Med. – High	Med.	Low	Med. – High
Accuracy: S; C (%)	-	81.8; 76.2	100; 88.3	-	100; 97.8
Average time: S; M (ms)	-	-	-	-	3971; 62183

Note: S: simple activity, M: mixed activities

3.6. Summary and Future Work

This chapter contributes to the knowledge in AR by developing a semantical segmentation approach which incorporates generic and user-preferred actions for a given ADL for future data analysis in AR process and provides context-aware and personalised assistance to the user. Additionally, a semantical knowledge modelling approach is developed which conceptualised generic knowledge as an ontological model and inhabitant specific preferences to conduct a particular ADL as asserted individual. Moreover, a semantical segmentation algorithm is designed to take individual sensor events upon activation and the knowledge model as input with multithreading processing to separate events into different ADL threads. Each ADL thread relies on a two-stage decision engine to find any association with the observed sensor event. In the first stage, the decision engine conducts T-box reasoning with generic KB and then A-box reasoning with observed sensor event and inhabitant specific preferences by querying the triplestore in the second stage. The second stage of decision engine is only invoked when the use of entity on which observed sensor is attached to has a contradiction or not been explicitly specified in generic ADL description. The ADL thread discards the observed event when the decision engine has failed to find any relationship. When the whole set of active ADL threads fail to see any relevance for a given sensor event, the start of a new ADL is assumed, and a new thread is created. The approach leverages the incremental Pellet reasoner, OWL & Jena API, and the notion of multithreading implementations techniques and tools to develop the semantical segmentation decision engine.

The proposed method was developed and evaluated against 30 test scenarios. The results indicate an improvement in segmentation accuracy compared to the counterpart studies with 100% and 97.8% for single and mixed activities scenarios with an average time of 3971ms and 62183ms. The main bottlenecks for high processing time are the synchronised incremental reasoning and duplicate ADL threads creation which ultimately created additional reasoning tasks and slowed down the overall process on the machine, which was limited to two cores.

Based on these findings, a future study is proposed to investigate in improving the segmentation performance by adapting Fork/Join parallelism framework [165] to efficiently split and manage tasks over multiple cores machine and utilise graphical processing unit (GPU). Moreover, investigating methods to support incremental Pellet reasoner thread-safe and parallel processing can encourage more real-time scalable solutions to emerge. Finally, focusing on comparing other segmentation approaches, developing accurate, fine-grained action level AR and learning algorithms with the support of the rule and temporal reasoning.

CHAPTER 4. FINE-GRAINED MIXED ACTIVITIES RECOGNITION

Limited physical mobility and forgetfulness are some of the common problems reported in the ageing population. Recent studies related to HAR within AAL systems made significant progress in recognising single-user activities and their actions at a coarse-grained level. However, limited studies have explored knowing user actions at a fine-grained level for a given activity using ambiguous sensor data measurements. Therefore, this chapter develops a fine-grained AR approach that uses semantic knowledge, fuzzy modelling and reasoning technique to recognise fine-grained level user actions for ADLs conducted in a smart environment. Fuzzy set theory is the core component of fuzzy modelling and reasoning to handle imprecise sensors data and fuse multimodal and more than one sensor information to improve the accuracy of AR results. The proposed approach has been evaluated in a microservices-based system using over 13,000 sensor events over two days from 19 individual sensors attached to 6 everyday objects within a real-time sensing environment. A set of 30 and 153 fuzzy rules were created to infer different states of the user performing two fine-grained actions. The results indicate that an average accuracy of 83.33% and 100% was achieved with a reasonable defuzzification duration for the two fine-grained actions.

4.1. Introduction

Dementia is frequently reported amongst the growing ageing population around the world [30], [31]. The common forms of Dementia are Alzheimer and Parkinson disease with several symptoms that hider the elderly’s ability to conduct ADL independently. Some of the key symptoms include memory loss, poor judgement, confusion, disorientation, hallucination, delusions and involuntary muscle contractions[31]. Therefore, AAL systems are being developed with the consideration of these symptoms to provide timely assistance to the users and notification services to the carer when required [166]. Wide-ranging work has been carried out to recognise single-user mixed activities scenario at a coarse-grained action level in the past decade. Yet, challenges in recognising activities with mixed action sequences at the fine-grained action level remain uncharted[167]. An example of mixed action sequences is when a user collects an object but abandon/forgets to use that object due to a fading memory or dropping/spilling the contents because of tremors.

As discussed in the previous section 2.1.1, the key variances in AR at coarse- and fine-grained action level are analysis of relationships and fusion of pieces of evidence, respectively. Figure 4.1 presents an example of *MakeTea*, *MakeToast* and *MakeBakedBeans* ADLs and their actions at two granularity levels. At the coarse-grained AR level, parameters such as context (i.e., time interval and location), relations between ADL_n descriptions and user's actions with everyday objects are used to deduce unfolding ADLs. In contrast, the fine-grained AR level studies inspect each action with everyday objects for a specific ADL to verify that the intended action has been conducted or left incomplete. Conditional to the required levels of monitoring the user's actions. Atomic actions are those who cannot be further decomposed. Such example of atomic actions can be "*filling*" kettle from the water tap, "*pouring*" water from the kettle into a cup and "*drinking*" from the cup action when conducting *MakeTea* activity.

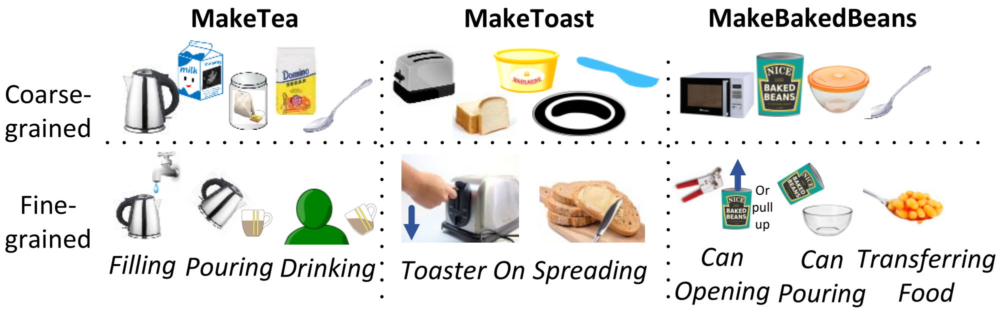


Figure 4.1. Coarse- and fine-grained granularity levels for three ADLs

The key research challenges in developing fine-grained action level AR are to model, collect multimodal sensors attached to everyday objects and fuse pieces of evidence adequately to verify and predict the completion of action accurately. Further issues are raised with non-binary sensor output data collected from the smart environment that is imprecise/vague and subject to interpretation [168], [169]. Subsequently, conceptualising and representing such ambiguous knowledge in a model and progressively reasoning with the incoming sensor data stream creates further action challenges. Hence, this chapter contributes by presenting an AR approach that detects fine-grained actions at the atomic level by collating pieces of evidence from multimodal sensors attached to everyday objects. Furthermore, enabling ambiguous non-binary sensor data to be interpreted and reasoned in a knowledge model and decision engine. The application of this approach can be seen in other research fields such as healthcare [170], [171], security, automotive [172], and energy management.

In the following sections, recent studies carried out to address fine-grained AR challenges are initially reviewed in section 4.2. Based on the gaps identified in the existing approaches, a novel approach and algorithm are proposed in section 4.3. Next, the system implementation details and evaluation results are provided in section 4.4 and 4.5, respectively. Section 4.6 concludes this chapter and discusses future research directions.

4.2. Related Work

Several data-driven (DD) and knowledge-based (KB) [20], [34] approaches have attempted to develop fine-grained action level AR by integrating imprecise/fuzzy sensor values[168], [169] and/or uncertainties[169]. These approaches frequently adapt mathematical theories such as fuzzy [166], [168], [172], probabilistic [37], [173], [174], possibilistic [175], [176], and Dempster-Shafer (DF)/Evidential theory[169] to model and reason with multimodal sensor data. In the following sections, studies are reviewed related to DD and KB approaches for AR at a fine-grained level in sections 4.2.1 and 4.2.2.

4.2.1. Fine-grained Level AR with DD Approaches

To achieve fine-grained AR with DD methods, work in [177] combined acceleration, acoustic and multi-sensor classifiers. These classifiers are J48 decision tree, random forest (RF), a Bayesian network, and support vector machine (SVM). A single off-the-shelf smartwatch was used in the experiment to recognise five daily activities, i.e., eating, vacuuming, sleeping, showering and watching TV. The result indicates that the combined approach achieved greater accuracy (91.5%) in comparison to individual classifiers. The main shortcoming of this approach is that the measurement of the sensors is not compared against the degree of use (imprecise). For instance, what is the “*minimum*” wrist movement required to infer the vacuuming action and if the watch is left on the table facing “*up*” before going to the shower compared to falling asleep with folded hands.

Work in [170], adopts weight-based probabilistic and conditional random field (CRF) decision classifiers with multimodal and multi-positional (wrist, back, leg and waist) sensors to achieve 80% AR accuracy of 19 coarse-grained and fine-grained routines in daily living. Likewise, work in [178] used an inertial ring and a bracelet to achieve fine-grained occupant activity recognition based on the wrist and index finger gestures of eating, drinking and brushing with favourable initial results. The limitations for both approaches are the ability to automatically link wearable sensors on the body part with gestures and embedded sensors with everyday objects. Furthermore, the inherently obtrusive nature and limited battery lifespan of wearable sensors create challenges for the widespread adoption of the system. Consequently, work in [179] developed a passive RFID based Moo Tag with onboard 3-axes accelerator sensor to attached to non-/perishable objects with ultra-high frequency RFID reader to detect fine-grained user action. The tag ID, Received Signal Strength Indicator (RSSI) and accelerometer values from the passive sensor tags are in congestion with HMM model to infer fine-grained actions. Nevertheless, these passive Moo Tag have limited computational, data storage and transportation capacity to attach more sensor data to increase the accuracy to determine completion of an action.

Work in [175], explored knowledge-based possibilistic network classifiers to handle uncertainty (imprecise, incomplete, missing) in sensor data when taking medication (with get water and take the pill as fine-grained actions) in AAL setting. Though, this approach still assumes that interaction with the everyday object as part of key sub-/action is the satisfactory complication of action. For instance, getting a cup and turning the tap on does not always mean the cup is being filled or "*minimum*" quantity of water is filled in the cup correctly. Therefore, additional sensors such as liquid level, accelerometer and gyroscope are required in the cup to be correlated and validate "*getting water action*". Additionally, limited support is shown to handle imprecise raw sensor data such as water level in the cup, and if the user has drank the water when detecting fine-grained action.

4.2.2. Imprecise Measurements with Knowledge-based AR

The knowledge-based approach initiates the modelling process by the formally conceptualising intricate knowledge by a domain expert(s). This knowledge model overcomes the "*cold start*" issue and increases reusability by modelling activities at multiple levels of abstraction. Nonetheless, the models created with knowledge engineering techniques require manual efforts[180], limited to the domain expert's knowledge, and incomplete.

In the KB approach, Web Ontology Language (OWL) is a backbone of semantic web language. OWL enables the formal representation of rich and complex knowledge by the domain expert(s) that can be reusable, human-readable and machine friendly. The ontology modelling techniques have been extensively leveraged to conceptualise concepts, describe relationships using a family of description logics (DLs) and reason with the explicitly defined information to deduce inexplicit knowledge. Yet, OWL and DL suffer from the ability to support imprecise/vague concepts.

An example of the study can be seen in [181], which presented a multi-level context-aware recognition framework(mlCAF). This framework developed a cross-domain (physical activity, nutrition, and clinical) ontological model and Web Rule Language (SWRL) rules-based reasoning. The low-level (fine-grained actions) contextual information such as nutritional and behavioural patterns of the inhabitants is initially inferred using cross-domain ontology-based inferencing with the support of the Pellet reasoner. The high-level context (coarse-grained activity) based on human behaviour and lifestyle is determined by using SWRL/SQRWL rules which keeps on making associations between three domains and low-level context at different levels. Similarly, work in [182] uses ontology and bespoke SPARQL Protocol and RDF query language (SPARQL) to recognise activities at two granularly levels. Nevertheless, both approaches suffer the same issue and add the complexity of manual querying.

The recent studies have extended the OWL expressivity capabilities and incorporated imprecise/vague concepts with the fuzzy ontology. The fuzzy ontology is based on fuzzy set theory. The fuzzy set theory allows one to associate a fuzzy concept with having a degree of membership in a given set by defining one value (Type-1) or two values between 0 and 1 (Type-2) [172]. Work in [166] developed a standalone Type-1 fuzzy logic system to recognise around 18 coarse-grained ADLs and human body motion. The types of sensors used in the system are physiological sensors, microphone, infrared sensors, debit sensors (for water flow) and state-change sensors. The system was developed using Labwindows CVI and C++ software. The preliminary results show that 97% accuracy in recognising ADLs. The fuzzy theory has been adopted to support decision making and combining multiple sensor data when recognising ADLs using ontology[183]–[185] and in other domains such as flight booking[172], and diabetic mellitus[171]. The common problems of these fuzzy ontology-based studies are the lack of emphasis on accurately detecting fine-grained actions based on object usage. Furthermore, there are limited tools available to develop fuzzy ontology and perform automatic reasoning. Though Umberto and his team have recently developed a fuzzy ontology plugin for Protégé [186] (ontology editor), and fuzzyDL[187] reasoner; see [188] more details. To the best of our knowledge, fuzzyDL plugin and reasoner have not been evaluated for detecting fine-grained AR within a real-time distributed system.

This chapter focuses on making four main contributions to recognising activities at the multi-granularity level. The first contribution is the approach to model coarse-grained and fine-grained actions required for ADLs using KD approach. The model at the coarse-grained action level consists of capturing complex context, environment, and relationships between everyday objects. Likewise, at the fine-grained actions level, everyday objects and their changing states are modelled with multimodal sensor (i.e., liquid level, temperature, accelerometer, and gyroscope) readings. The second contribution is the approach to represent the imprecise nature of some non-binary sensor readings into fuzzy concepts/state of a given object (i.e., kettle water temperature is "*hot*"). The third contribution is the approach to fusion multimodal sensor readings to detect fine-grained actions. For instance, pouring action for a kettle can be defined when the temperature is "*hot*", the liquid level is "*full*" and gyroscope Z value is "*tilt*"). The fourth contribution is the decision engine that progressively takes multimodal sensor readings and multi-granularity knowledge model as inputs to calculate the degree of action completion.

The decision engine algorithm has been developed and evaluated in a distributed prototype system.

4.3. Fuzzy based Fine-grained AR Approach

A KD based AR approach that combines multimodal sensor observations and its imprecise/vague values to detect actions at a coarse and fine-grained level is proposed. The goal of detecting activity at the coarse-grained level is to identify an object's relation to ADL description and sensor network. Three key types of satisfactory criteria of the ADLs are conceptualised to understand in the overall context: spatial (location), temporal (time interval) and critical actions (object interactions). For this, OWL-based semantical modelling and reasoning methods are applied to identify the relationships of the everyday object with the sensor network and ADLs description.

In contrast, fine-grained AR level detects in-/complete atomic action(s) with each object for a given activity. Each object is attached with one or more multimodal sensors to increase the accuracy of detecting the atomic action. The multimodal sensors produce non-binary data which creates an imprecision and vague interpretation of the object state. Therefore, fuzzy logic is leveraged to define a gradual range in which an object is in a given state. For instance, if the liquid level of a kettle is between 16.85- 17.47 picofarads (pF), then the kettle is half (medium) full. To fusion the multimodal sensors with non-/binary sensors, fuzzy rules are used to deduce the completion of an atomic action with a given object at a given time instance. For example, a "*pouring*" hot water atomic action rule can be defined when the liquid level is "*medium*" full, object temperature "*hot*" and "*medium*" tilt position (gyroscope Z-axis); for more details in creating fuzzy rules see [189]. Figure 4.2 presents a conceptual view of the fine-grained AR approach. The fine-grained AR approach consists of three main phases: (1) data segmentation, (2) windowing and (3) fine-grained AR.

The data segmentation phase (1) is responsible to semantically segment sensor data based on the relationship of what object the sensor is attached to and description of ADL. Section 4.3.3 provides more details on sensor data segmentation. The windowing phase (2) inspects sensor data for a given ADL within a fixed time interval window (W_n). The dynamic windowing mechanism has been presented in the past [27], and it is not the focus of this chapter. Nonetheless, the critical tasks in this windowing phase are to take minimum and maximum values of non/binary sensors data relevant to ADL_n within each W_n as input for fine-grained AR phase. These sensors input values are stored as instances for each object state at W_n time interval in the domain knowledge base (DKB) model.

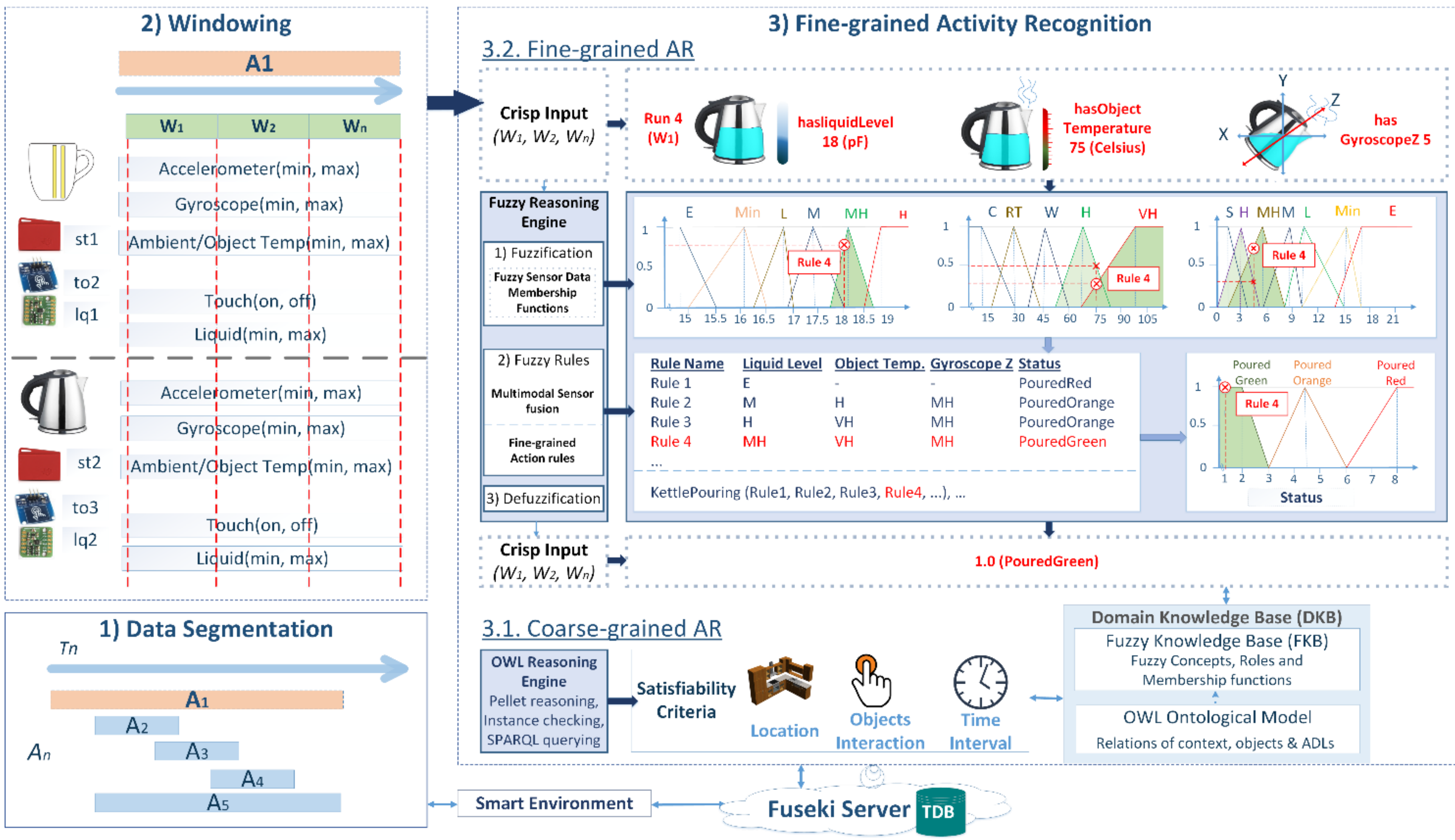


Figure 4.2. Overview of fine-grained AR approach in three phases: (1) data segmentation, (2) windowing and (3) fine-grained AR.

The fine-grained AR phase (3) consists of analysing instances of sensor input at each W_n using semantic (*3.1*) and a fuzzy reasoning engine (*3.2*). In phase 3 of Figure 4.2, a conceptual view of the approach is shown on the left and an example of W_n on the right. The semantic reasoning engine (*3.1*) utilises ontological model containing the ADL and context model (see in section 4.3.1.1) to identify the relationship between a sensor at the coarse-grained level. For this, the Pellet reasoner[190] and SPARQL[191] querying based approach are leveraged. In turn, the fuzzy engine detects fine-grained actions using multimodal sensors and fuzzy rules. The fuzzy engine (*3.2*) takes instances of W_n containing multimodal sensor data for each object as crisp input and fuzzy knowledge-based (FKB) model to produces crisp outputs. The fuzzy reasoning engine is composed of three main components, fuzzification, fuzzy rules to describe fine-grained actions and defuzzification method. Section 4.3.1.2 describes the FKB modelling process and the role it plays in the fuzzy reasoning engine.

An example of the fuzzy reasoning engine detecting the "*pouring*" action for A_1 from W_1 time interval containing the kettle's liquid, temperature and gyroscope sensors data is illustrated in Figure 4.2 (right of part *3.2*). The fuzzy reasoning engine first identifies the membership functions for the crisp sensor values, run fine-grained action rules based on objects and then produces defuzzification results. A fine-grained action can be said complete, unsure or incomplete in different scenarios. Hence, a combination of multimodal sensors and their states allow creating scenarios to be described as sub-rules. In this example, the *KettlePouring* action rule consists a set of sub-rules (*Rule 1*, …, *Rule N*) that details all possible scenarios in which the action is incomplete (*PouredRed*), unsure (*PouredOrange*) or complete (*PouredGreen*). The successful completion status (*PouredGreen*) scenario is described in *Rule 4* containing medium-high (MH) liquid level, very hot (VH) object temperature and tilt position MH. Likewise, *Rule 1* describes incomplete (*PouredRed*) and *Rule 2* as unsure (*PouredOrange*) scenarios. Based on the W_1 input values, *Rule 4* best matches the membership functions as indicated with a red cross and dashed line respective sensor inputs.

4.3.1. Crisp and Fuzzy Knowledge Modelling

4.3.1.1. ADL and Context Modelling

The knowledge base ($\mathcal{KB}$) is developed to conceptualise crisp (σ), and imprecise (π) sensor data measurements within an ontological model as denoted in equation 4-1.

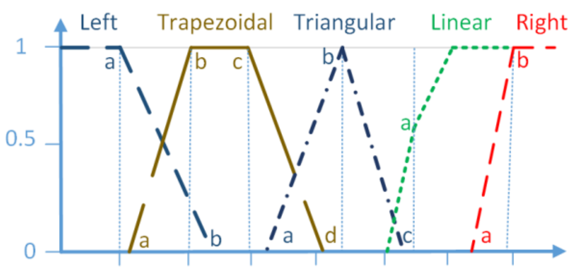
$$\mathcal{KB} = \{\sigma, \pi\} \qquad \textbf{4-1}$$

The crisp knowledge model comprises of description and relationships between ADLs ($\mathcal{ADL}_i$), the environment ($\mathcal{Env}_a$), and sensors network ($\mathcal{SN}_d$) as denoted in equation 4-2. $\mathcal{ADL}_i$ contains a set of activities ($activity_n$) and actions ($action_m$). Each $action_m$can be recognised at multi granularity level: coarse (ca) and fine-grained (fa). The ca actions involve recognising the relationship between objects and ADL. However, the fa analyse the set of sensors data produced by multiple sensors attached to an object to verify the intended action has been fulfilled. An $activity$ is performed in a given environment ($\mathcal{Env}_a$) by a one or more people ($\mathcal{P}_r$) at a given time interval (TI_s). Each $\mathcal{Env}_a$ has location (loc_b) information and set of objects (Obj_c) which is monitored by the sensing network ($\mathcal{SN}_d$). Each Obj_c can have one or more sensing platforms ($plat_e$) attached to them capturing different parameters of Obj_c. For this, each $plat_e$ can host one or more multimodal sensors (s_f) to analyse object interactions.

$$\begin{aligned} \sigma = \{&\mathcal{ADL}_i[activity_n\{action_m[ca, fa]\}], \\ &\mathcal{Env}_a\{loc_b, Obj_c\}, \\ &\mathcal{SN}_d\ \{Obj_c[plat_e\{s_f\}\ ],\ \mathcal{P}_r,\ TI_s\} \end{aligned} \qquad \textbf{4-2}$$

4.3.1.2. Imprecise Knowledge Modelling

The core element of fuzzy OWL is Fuzzy Logic. Fuzzy logic is based on the theory of fuzzy sets proposed by L. Zadeh[168] to support imprecise and vague knowledge. The fuzzy set theory enables imprecise sensor value to be assigned as a member of a given set with a membership degree between 0 and 1 for a Type-1 fuzzy set. In comparison to the classical set theory, elements are either part of a given set or not, i.e. 0 (false) or 1 (true). The Type-2 fuzzy set introduces secondary membership functions where upper, and lower membership boundaries are created when it is challenging to recognise simple fuzzy membership value for fuzzy terms/concepts. The region between the upper and lower membership boundaries is called the footprint of uncertainty[172]. The fuzzy ontology allows Type-1 and Type-2 fuzzy sets membership boundaries to be defined using `Datatype` annotations and `minValue/maxValue` attributes[192].



d → left (k1, k2, a, b)

right (k1, k2, a, b)

triangular (k1, k2, a, b, c)

trapezoidal (k1, k2, a, b, c, d)

mod (*d*)

mod → linear (c), triangular (a, b, c)

(k1 = minimum, k2 = maximum values)

Figure 4.3. Type-1 fuzzy membership functions and modifiers

FKB formally conceptualises imprecise sensor data and fusion of multiple sensors as rules to determine the completion of a given fine-grained action to satisfactory thresholds. There are three critical steps in developing a fuzzy ontology; (1) fuzzification, (2) rules and inferring system, and (3) defuzzification.

In the fuzzification step, the vague sensor data sets are described as fuzzy concepts ($\mathcal{FC}_b$) with fuzzy membership functions (*d*) and modifiers (*mod*) defined in Figure 4.3. The membership functions are trapezoidal, triangular, left(-shoulder), right(-shoulder), crisp interval, and linear. The modifiers are linear and triangular. Table 4.1 presents a fragment of the three types of imprecise sensor data as a Type-I fuzzy concept; temperature, liquid and position (IMU) data. The fusion of multimodal sensor data attached to a given object is considered to increase the accuracy of the fine-grained kettle “*pouring*” hot water action detection. The temperature sensor values are associated with linguistical concepts such as “*hot*”, “*cold*” and “*normal*” which is often subjective to a given context or person. Similarly, the liquid level sensor enables one to categorise if a container in varying size/dimension is “*full*”, “*half full*” or “*empty*”. Whereas, IMU sensors (i.e., accelerometer, gyroscope) enables one to understand the position of the object and how it has moved in three-dimensional space. Therefore, combining these three parameters, fuzzy rules can be created to define how much one needs to tilt the container to “*pour*” hot water into another container with respect to the liquid level. Likewise, other fine-grained actions defined in FKB are “*filling up*”, and “*drinking*” from the container (i.e., cup or kettle) can be.

The fuzzy rules are constructed mainly with Mamdani and Takagi/Sugeno approaches[166]. The fuzzy rules are constructed with IF (antecedent) and THEN (consequent) statements. Table 4.2 illustrates partial fuzzy rules for a kettle to infer pouring state based on the liquid level, object temperature, and gyroscope Z-axis position. Table 4.2 presents three sets of rules specifying scenarios in which pouring action is incomplete, unsure and complete with respective *PouredRed, PouredOrange, PouredGreen* flags. The first set of rules are for incomplete pouring action scenarios. The rule, `rule_kettle_empty`, states that if the kettle’s liquid level is empty (`some liquidLevel kettle_Liquid_Empty_ls`) then poured status flag is red (*some pouredStatus PouredRed*). Likewise, the second set of rules define two scenarios where it is unsure if the pouring action has been completed. First rule, `rule_kettle_objTemp_warm`, states that if kettle’s object temperature is warm then poured status flag is orange. The second rule, `rule_high_veryHot_water_zhigh_midHigh`, state if the liquid level is high (`some hasLiquidLevel kettle_Liquid_High_rs`), object temperature is warm, and tilt threshold is medium to high (`some hasAccelerationZ kettle_gyro_z_pour_thres_liquid_midHigh_tri`) then the poured status is orange (`some pouredstatus PouredOrange`).

Table 4.1. Fragment of Fuzzy Concepts, Roles, and Membership functions for Multimodal Kettle "*Pouring*" Action Rules in FuzzyDL Syntax

```
% 1) Membership functions for fuzzy concepts
(define-fuzzy-concept kettle_Liquid_Empty_ls left-shoulder (0.0, 100.0, 14.47, 15.5))
(define-fuzzy-concept kettle_Liquid_Minimum_tri triangular (0.0, 100.0, 15.0, 16.25,
16.5))
(define-fuzzy-concept kettle_Liquid_Low_tri triangular (0.0, 100.0, 16.35, 16.83,
17.0))
(define-fuzzy-concept kettle_Liquid_Medium_tri triangular (0.0, 100.0, 16.85, 17.47,
18.0))
(define-fuzzy-concept kettle_Liquid_MediumHigh_tri triangular (0.0, 100.0, 17.75,
18.12, 18.6))
(define-fuzzy-concept kettle_Liquid_High_rs right-shoulder (0.0, 100.0, 18.5, 18.68))
(define-fuzzy-concept kettle_objTemp_veryhot_rs right-shoulder (-150.0, 150.0, 75.0,
100.0))
(define-fuzzy-concept kettle_objTemp_hot_tri triangular (-150.0, 150.0, 50.0, 70.0,
80.0))
(define-fuzzy-concept kettle_objTemp_warm_tri triangular (-150.0, 150.0, 35.0, 45.0,
60.0))
(define-fuzzy-concept kettle_gyro_z_pour_thres_liquid_ min_tri triangular (-50.0,
50.0, 10.0, 15.0, 17.0))
(define-fuzzy-concept kettle_gyro_z_pour_thres_liquid_ midHigh_tri triangular (-50.0,
50.0, 2.0, 5.0, 8.0))
(define-fuzzy-concept PouredGreen left-shoulder (0,9,1,3))
(define-fuzzy-concept PouredOrange triangular (0,9,3,4.5,6))
(define-fuzzy-concept PouredRed right-shoulder (0,9,8,9)) …
% Fuzzy Relationships between Concepts
(functional hasObjectTemperature)
(range hasObjectTemperature *real* -150 150)
(functional hasGyroscopeZ)
(range hasGyroscopeZ *real* -100 100)
(functional pouredstatus)
(range pouredstatus *real* 0 9) …
```

Table 4.2. Partial Kettle "*Pouring*" description using FuzzyDL Rules

```
% 2) Multimodal sensor rules for kettle pouring action
% 2.1) PouredRed - pouring incomplete rules
(define-concept rule kettle empty (g-and (some hasLiquidLevel kettle_Liquid_Empty_ls)
(some pouredstatus PouredRed))) …
% 2.3) PouredOrange – pouring potentially completed
(define-concept rule_kettle_objTemp_warm (g-and (some hasObjectTemperature
kettle_objTemp_warm_tri) (some pouredstatus PouredOrange)))
(define-concept rule_high_veryHot_water_zhigh_midHigh (g-and (some hasLiquidLevel
kettle_Liquid_High_rs) (some hasObjectTemperature kettle_objTemp_veryhot_rs) (some
hasGyroscopeZ kettle_gyro_z_pour_thres_liquid_midHigh_tri) (some pouredstatus
PouredOrange))) …
% 2.4) PouredGreen – pouring successfully completed
(define-concept rule midHigh veryHot water (g-and (some hasLiquidLevel
kettle_Liquid_MediumHigh_tri) (some hasObjectTemperature kettle_objTemp_veryhot_rs)
(some hasGyroscopeZ kettle_gyro_z_pour_thres_liquid_midHigh_tri) (some pouredstatus
PouredGreen))) …
% 2.5) Combining all kettle sensor states
(define-concept rulePOURING (g-or rule kettle empty rule kettle objTemp warm
rule high veryHot water zhigh midHigh rule_midHigh_veryHot_water …)
```

Table 4.3. Example of MOM defuzzification Query results of pouring rules in four scenarios with Multimodal sensors input data from the Kettle

% 3) Input – We consider four scenarios (S1-S4)	
%S1)liquidLevel=empty, objTemp=hot, gyroZ=static (instance run1 (= hasLiquidLevel 10)) (instance run1 (= hasObjectTemperature 80)) (instance run1 (= hasGyroscopeZ 2)) *% output ==> pouredstatus = 9.0 (red)* *%S2)liquidLevel=empty, objTemp=warm, gyroZ=static* (instance run2 (= hasLiquidLevel 10)) (instance run2 (= hasObjectTemperature 45)) (instance run2 (= hasGyroscopeZ 2)) *% output ==> pouredstatus = 6.75 (orange)*	*%S3)liquidLevel=high, objTemp=veryhot, gyroZ=midhigh* (instance run3 (= hasLiquidLevel 20)) (instance run3 (= hasObjectTemperature 75)) (instance run3 (= hasGyroscopeZ 5)) *% output ==> pouredstatus = 4.5 (orange)* *%S4)liquidLevel=midhigh,objTemp=veryhot, gyroZ=midhigh* (instance run4 (= hasLiquidLevel 18)) (instance run4 (= hasObjectTemperature 75)) (instance run4 (= hasGyroscopeZ 5)) *% output ==> pouredstatus = 1.0 (green)*

Equally, the completion rule, *rule_midHigh_veryHot_water*, states that if the kettle's liquid level is medium to high (*some hasLiquidLevel kettle_Liquid_MediumHigh_tri*), object temperature is very hot (*some hasObjectTemperature kettle_objTemp_veryhot_rs*), and minimum threshold tilt degree registered (*some hasAccelerationZ kettle_gyro_z_pour_thres_liquid_midHigh_tri*), then poured status is green (*some pouredstatus PouredGreen*). Lastly, all the other possible combinations of the sensor status and scenarios are added to the main pouring rule (*rulePOURING*) concept to determine which rule is best matched for a given sensor input in the defuzzification step.

The final defuzzification step consists of using the sensor input values and fuzzy rules to query a membership value for a given action. The conventional defuzzification methods available are Centroid Of Area (COA), Bisector Of Area (BOA), Mean Of Maximum (MOM), Smallest Of Maximum (SOM) and Largest Of Maximum (LOM)[166].

Figure 4.4 illustrates MOM defuzzification results of the four scenarios with the multisensory data of a kettle and whether the pouring task is incomplete (S1), unsure if completed (S2-S3) or completed(S4). In the scenario S1, instance of kettle containing values of liquid level 10pF (empty), object temperature 80Celsius (very hot) and gyroscope z-axis value to be 2 (the threshold for pouring when liquid level medium) has been defined with *pouredStatus* MOM defuzzification output as 9.0 (*PouredRed*). Equally, in S2, change of object temperature to 45 Celsius resulted in MOM defuzzification value to be 6.75 (*PouredOrange*). On the other hand, scenario S3 containing liquid level 20 (high), object temperature 75 Celsius (*very hot*) and gyroscope z-axis value 5 (the threshold for pouring when the liquid level is `medium-high`) resulted in 4.5 (`PouredOrange`). Likewise, the last scenario S4 result in 1.0 (`PouredGreen`) as the kettle's liquid level is 18pF (*medium-high*), object temperature is 75 (`very hot`), and the gyroscope Z axis is 5 (the threshold for pouring when the liquid level is *medium-high*).

4.3.2. Multimodal Sensing Attributes

The fusion of ambient and dense multimodal sensing environments is proposed to detect coarse- and fine-grained actions. The ambient sensors provide coarse-grained contextual information about the environment and the objects which users interact with, i.e., multi-sensor with a motion detector, door/window opening and closing magnetic sensors. In contrast, dense sensors such as TI SensorTags for object positioning and liquid level sensing approach are proposed to be attached to the relevant everyday objects for fine-grained object usage recognition. For instance, "*pouring*" water from the kettle to a cup can be determined if the correlation between the changing state of the water level and tilting position of the kettle and cup exceeding a given threshold. This threshold can vary depending on the initial quantity of the water level, dimensions and the sensor placement on the kettle. Figure 4.4 depicts the overall sensing data types for coarse-/fine-grained AR.

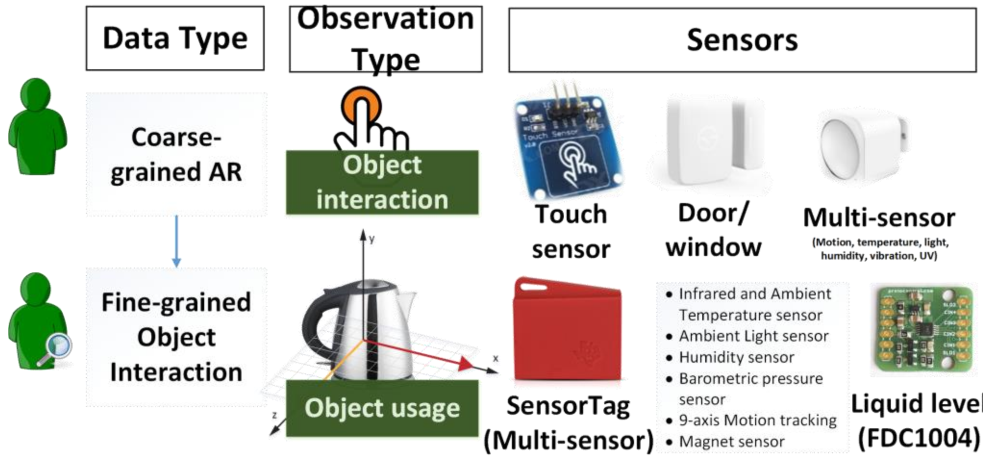


Figure 4.4. Proposed sensing parameters for coarse-/fine-grained ADL detection.

4.3.3. Sensor Data Segmentation

The segmentation process inspects each sensor event incrementally, twofold. Firstly, terminology (T-box) based reasoning is performed on OWL classes to check if the given event is part of an ongoing candidate ADL class, otherwise, it creates a new activity queue for the first event. These checks involve performing satisfiability of the concept, subsumption of concepts, and instance checking using incremental Pellet reasoner. The second step is only executed if there are any conflicts identified in step one. In the second step, assertion-based (A-box) reasoning is performed on class instances by querying the triplestore to find relevant ADL preferences specified by the user. In the case where both steps fail to see any association of the sensor and object with ongoing activity, the start of the new activity is assumed. For this purpose, the notion of multithreading is used where each thread represents individual ongoing ADL, and these ADL threads capture any sensor events relevant to that activity. In addition, session manager and event recycler threads run in parallel with these ADL threads for housekeeping tasks. The session manager thread checks the left-over sensor observations and creates a new ADL thread, and the event recycler thread maintains the sensor observations queue. The comprehensive details on how two types of knowledge are modelled and used for the semantical segmentation can be found in CHAPTER 3.

4.3.4. Fine-grained AR Algorithm

Table 4.4 presents the algorithm as a pseudo-code of the processing thread (PT_x) of an ADL_n that performs AR at the coarse and fine-grained action level. The algorithm takes in segmented sensors (*segmentedSensors*) and the candidate ADL class (*adlClass*) based on T-Box reasoning from the segmentation process as inputs. The reasoning engine outputs the results (*arResult*) of OWL (line 10) and fuzzy reasoner (line 25). The algorithm conducts three main stages.

Table 4.4. Pseudocode for Fine-Grained AR in ADL Processing Thread (PT_x)

```
ALGORITHM 1: Input: segmentedSensors, adlClass, from, to
             Output: arResult
1  List<String> loc = getLocations(adlClass);
2  List<String> obj = getObjects(adlClass);
3  List<Long> range = getIntervals(adlClass);
4  Result arResult = new Result();
5  //1) CONTEXT ANALYSIS (Coarse-grained AR)
6  for (Sensor s: segmentedSensors)
7     updateList(loc, getLocations(s));
8     updateList(obj, getObjects(s));
9     updateList(range, s.getTimestamp())); endfor
10 arResult.addCoarseResult(loc, obj, range);
11 //2) PREPARE (Fine-grained AR)
12 List<String> fga= getALLFinegrainedActions(adlClass);
13 if (fga.size()>0)
14    List ws = getSensorDataBetween(segmentedSensors , from, to);
15    Map<String, List> mapWs = calculateMinMax(ws);
16    FParams objDataInstance = populateFKB(mapWs, from, to); endif
17 //3) DETECTING IN-/COMPLETE ACTIONS (Fine-grained AR)
18 for (Param p: objDataInstance.getParms())
19    FuzzyyResult fr= FuzzyDLUtils.run(p.getRule(), p.getProperty(),p.getInstanceName());
20    arResult.addFuzzyResult(fr); endfor
21 storeInTDB(arResult); return arResult; // output/store in TDB
```

The first stage (lines 1-11) involves performing coarse-grained level AR, where contextual satisfactory attributes from the OWL model are retrieved for *adlClass*. These contextual satisfactory attributes include L, TI, and KO. These attributes are retrieved from performing SPARQL querying using `getLocations`, `getObjects` and *getIntervals* functions respectively (lines 1-3). A full and temporary list for these attributes is maintained by the `updateList` function when analysing each sensor input from the `segmentedSensors` list (lines 6-9). The analysis of each sensor consists of identifying the relationship between the object hosting the sensor and its contextual attributes. The temporary list removes the attributes once they have been identified to show missing information from the sensor data.

The second stage (lines 11-17) consists of preparing the FKB model for fuzzy-based AR in stage three. The preparation stage involves performing three steps. The first step is to retrieve all fine-grained actions (line 14) from FKB based on the candidate ADL class (`adlClass`). The second step is to calculate minimum and maximum values (line 15) of each of the sensor data within a fixed window size. The third step is to store all the instances of objects and the hosted sensors data at the time window (W_n) in the *fuzzyDL* syntax. Furthermore, the fine-grained actions rules and data properties are mapped with each everyday object data instance to the `objDataInstance` (an instance of `FParams` class) by the `populateFKB` function (line 16).

The third stage (lines 18-21) include performing fine-grained action detection using the fuzzy reasoning engine. The fuzzy reasoning engine requires multimodal sensor data (msd_m) of each object (Obj_k) within a fixed W_n as inputs for the fuzzy reasoning engine. In addition, the fuzzy engine requires FKB containing fuzzy membership (FM_a) functions of fuzzy concepts (FC_b) and fine-grained action rules (FAR_c) to perform defuzzification. The defuzzification is performed based on fine-grained action fuzzy rules relevant to everyday objects defined in *objDataInstance*. Therefore, enabling the fuzzy reasoning engine to perform defuzzification (i.e., using MOM method) on a small set of rules (line 19). All the defuzzification query results of the are stored in the `arResult` instance on line 20, triplestore and publish results to clients on line 21.

4.4. System Implementation

This fine-grained AR approach is developed in a microservice architecture (MSA) system illustrated in see Figure 4.5. The MSA has been developed using lightweight REST-based communication protocol. The system was predominately built using the Java programming language. The MSA improve the interoperability, scalability and performance of the system. The external clients make requests to a single web service, SmartWeb API. The SmartWeb API web service liaises with four internal web services to route the client’s requests to relevant web service(s). These four internal web services are: application API, service API, sensing platform API and data storage API.

The application API enables users to manage their profiles and receive ADL assistance. The user profile management feature allows details such as the inhabitant's details, and ADL preferences record to be stored using data storage API. The main requirements of data storage API are to store, update and retrieve time-series based sensor events log, AR results, inhabitant's ADL preferences and other application-specific records. The critical role of the ADL assistance tasks is to provide just-in-time assistance based on AR results produced by service API.

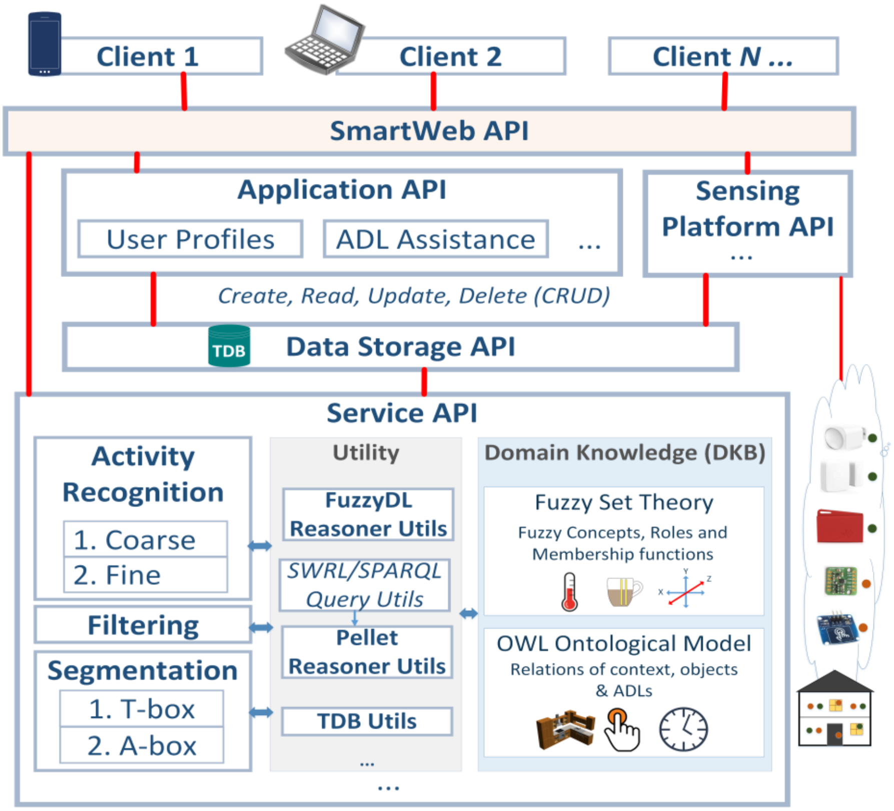


Figure 4.5. Fine-grained AR System with Service-oriented Architecture.

The service API is the core component of the AAL system. The ADL assistance feature in the application API relies on the service API to analyse the sensing data received from the sensing platform API. The service API perform three tasks: data segmentation, data filtering and activity recognition with the reasoning tools. The segmentation approach discussed in CHAPTER 3 and elaborated in section 4.3.3 is developed to separate and group sensor observations. The segmentation approach utilises the semantic relationship between sensors, objects, ADL descriptions and user-specific preferences knowledge to group sensors data. The second task of service API is to handle the errors in sensor measurements such as drift in accelerometer and gyroscope over time. Hence, complementary and Kalman filtering techniques are commonly applied for filtering and smoothing the drifted data before performing fine-grained AR algorithm. The sensing platform API collects data from the smart environment, store in Jena Fuseki triplestore (TDB) and broadcast the events to clients using server-side event (SSE) messaging protocol.

4.4.1. OWL and Fuzzy Knowledge Modelling

The ontology editing tool Protégé has been leveraged to describe generic everyday objects within the living environment and their relationship with ADLs and sensing platforms. Figure 4.6 illustrates a hierarchical conceptualisation of environmental objects, sensors, and the location as OWL classes. These classes enable ADL to be intricately described using OWL's capabilities, such as the fragment depicting *MakeTea* activity in Figure 4.7. The *MakeTea* fragment describes the activity relationship between everyday objects required to complete the activity with some cardinality restrictions. Figure 4.8(a) provides a screenshot of fuzzy membership function being created for imprecise sensor data types. Figure 4.8 (b) shows a fragment of fuzzy rules for fine-grained "*pouring*" action of the kettle in fuzzyDL syntax.

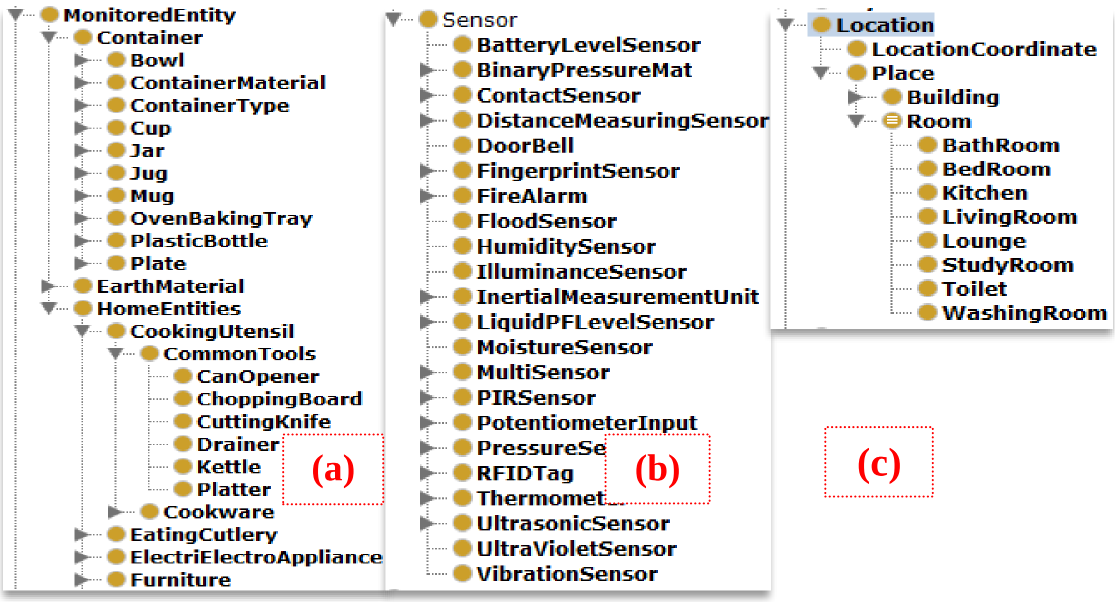


Figure 4.6. A fragment of the ontology describing everyday objects (a), sensors (b) and location (c).

4.4.2. Reasoning Tools and Storage

The incremental pellet reasoner [190] and SPARQL[191] querying approach is used to perform semantical segmentation and fine-grained AR. SPARQL is used to query TDB for deriving the relationship between sensor event, an object that hosts the sensor. Whereas, the incremental pellet reasoner is used to perform subsumption and instance checking with ADLs description in OWL model. Other tools used are fuzzy ontology editor plugin for Protégé [186] and fuzzyDL[187] reasoner. The plugin was used to model imprecise sensor data and fine-grained actions and rules. The fuzzyDL reasoner is then used to parse fuzzy ontology file (OWL) into fuzzyDL syntax and perform defuzzification queries.

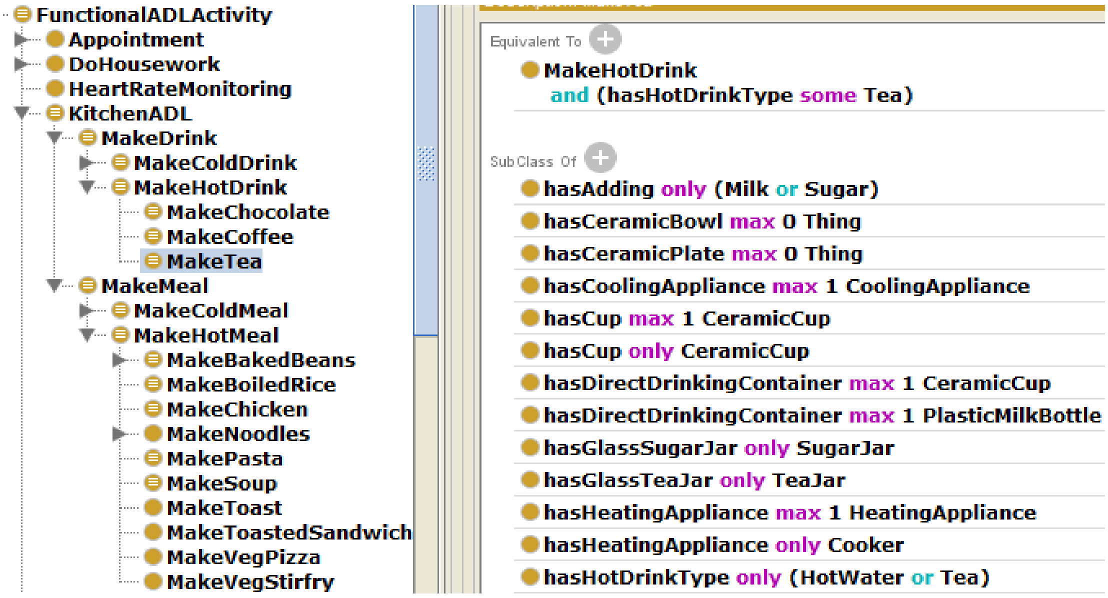


Figure 4.7. A fragment of *MakeTea* activity description with the relationship between everyday objects, mandatory and optional actions.

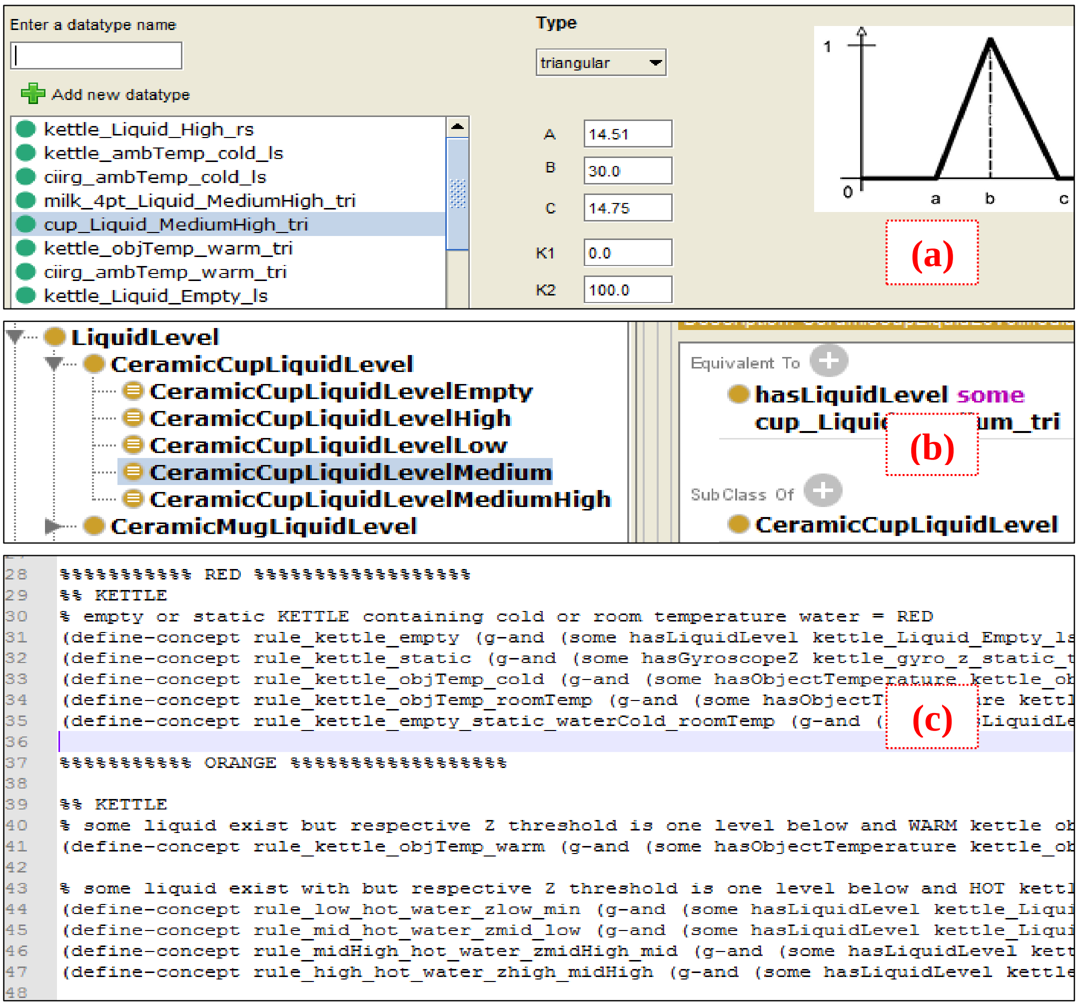


Figure 4.8. (a) Fuzzy OWL plugin, (b) fuzzy concepts, and (c) Fuzzy DL rules using a text editor.

The service API contains a *Utility* layer containing several supporting Java classes such as *fuzzyDLReasonerUtils*, *PelletReasonerUtils*, and *TDBUtils*. The *fuzzyDLReasonerUtils* Java class maintain FKB, take sensor data as input and perform defuzzification queries based on fuzzy rules. Whereas, the *PelletReasonerUtils* and *TDBUtils* interact with OWL model files and TDB. TDB stores and updates record such as the knowledge model, user's ADL preferences, sensor events log, and AR reasoning results. The results are exposed to client devices via *SmartWebAPI* layer using RESTful communication protocol and JavaScript Object Notation (JSON) data format. The java library JAX-RS API [193] was leveraged to develop RESTful web service. More details of system architecture and the hardware sensing configuration can be seen in previous work [158].

4.4.3. System Interface

The web interface is developed using multiple design java scripts such as material and angular. The data visualisation javascript libraries such as D3, list.js and vis.js are employed to enable the user to view multimodal sensor data interactively. Figure 4.9 shows three fragments of the web browser interface. Figure 4.9 (a) depicts the activity recognition and associate sensor events to the three activities. Figure 4.9 (b) presents a visualisation of multiple types of sensors attached to the kettle.

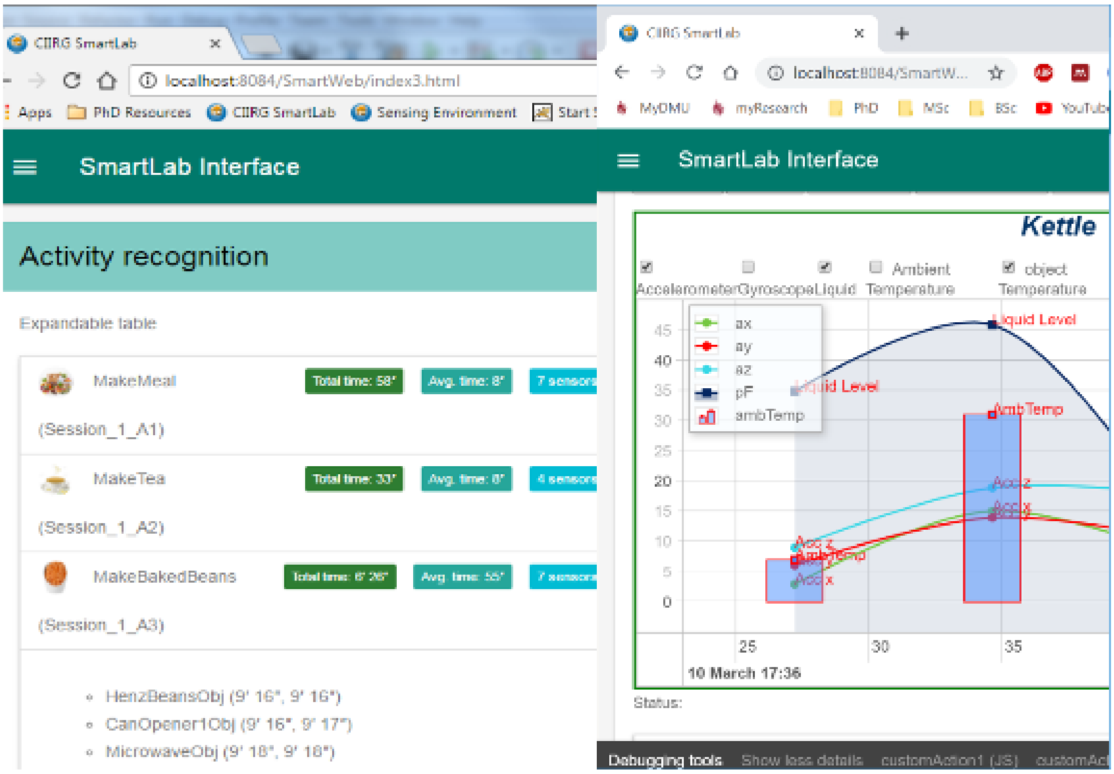


Figure 4.9. Web interface: (a) activity recognition page, (b) display multimodal sensors attached to the kettle

4.5. Evaluation

4.5.1. Experiment design

MSA and fine-grained AR algorithm are evaluated by collecting the dataset initially for two kettle based fine-grained actions: *filling* (F1), *pouring* (F2) for *MakeTea(A1)* activity. A1 activity with F1 and F2 actions was carried under two test scenarios where the actions are fully completed (T1) and partially completed or missing (T2). A sum of 6 everyday objects with at least 18 individual sensors and up to 8 types of sensors were attached to everyday objects to create a dataset as detailed in Table 4.5. The sampling rate of the continuous sensors (‘) is 500ms and a fixed sliding window size of 5s is used to analyse the multimodal data. The raw data values were then used to create fuzzy membership functions for each imprecise sensor parameter types and create fuzzy set rules accordingly.

Table 4.5. Everyday objects associated with Three ADLs and Nine fine-grained actions.

Sensor type/ Activity & Objects	**Arduino**				**Sensor Tag**					**Securefi**
	AB ID	*T*	*L’*	*PO*	*ST ID*	*A’*	*G’*	*AT’*	*OT’*	*D/W*
A1 – MAKE TEA, + Filling Kettle (F1), Pouring to Cup (F2), Adding Sugar(F3), Stir (F4)										
Cup	1	✓	✓		1	✓	✓	✓	✓	
Kettle	2	✓	✓		2	✓	✓	✓	✓	
Water Tap	4	✓		✓						
Tea/Jar	5	✓✓								
Sugar/Jar	6	✓✓								
Spoon1					6	✓	✓			✓

Note: {ABID: Arduino board ID, T: touch, L: liquid, PO: potentiometer}, {STID: sensor tag ID, A: accelerometer, G: gyroscope, AT: ambient temperature., OT: object temperature}, {D/W: door/window}. +: fine-grained actions, **’**: continues sampling sensor

Next, the accuracy and performance of the fuzzy-based fine-grained AR algorithm are evaluated by comparing the MOM defuzzification results against ground truth and duration of time for the calculation. Thus, eliminating factors such as network delays, communication errors, and time synchronisation errors from the experiment. The input from multimodal sensors attached to the object at six different time intervals (*TI*) and a set of fuzzy rules were provided to the fuzzyDL reasoner to recognise the actions. The fuzzy rules set consists of 30 and 153 possible scenarios in which F1 and F2 action states can be recognised. Each fuzzy rule can contain more than two types of sensor fuzzy membership states to determine if an action is complete {0-3}, unsure {3-6} or incomplete {8-9}. The experiment was repeated three times to measure the average duration for recognising two actions at six different time intervals.

The proposed MSA based system was developed with two machines running multiple web services to collect and analyse the data. The SmartWeb API and application API web services were running on a windows 10 laptop. The hardware configuration of the laptop is an i7 2.60GHz processor with 2-cores and 8GB RAM. The service API, sensing platform API and data storage API web services were running on a windows server 2012. The server configuration is an i5 3.00GHz processor, 8GB RAM and 4 cores.

Table 4.6. Accuracy and Performance Results for fine-grained AR Approach Within Single Activity Scenarios.

Run	**TI**	**F1**	**Type**			**F2**	**Type**		
		#		*Acc. %*	*Rules: 30 +ms*	*#*		*Acc. %*	*Rules: 153 +ms*
1	1	9.0	T2	100	23193	9.0	T2	100	346869
	2	9.0	T2	100	21553	9.0	T2	100	58755
	3	0.53	T1	100	21409	9.0	T2	100	58085
	4	0	T2	0	20803	9.0	T2	100	71112
	5	0.64	T1	100	20820	9.0	T2	100	59466
	6	0.67	T1	100	20890	8.92	T2	100	60603
2	1	9.0	T2	100	23810	9.0	T2	100	340594
	2	9.0	T2	100	23821	9.0	T2	100	62999
	3	0.53	T1	100	24812	9.0	T2	100	57808
	4	0	T2	0	26439	9.0	T2	100	58144
	5	0.64	T1	100	26074	9.0	T2	100	58048
	6	0.67	T1	100	25399	8.92	T2	100	58354
3	1	9.0	T2	100	30314	9.0	T2	100	321308
	2	9.0	T2	100	27074	9.0	T2	100	57882
	3	0.53	T1	100	28805	9.0	T2	100	55990
	4	0	T2	0	25377	9.0	T2	100	57396
	5	0.64	T1	100	26766	9.0	T2	100	57380
	6	0.67	T1	100	26297	8.92	T2	100	54922
Avg.				**83.33**	**24647.56**			**100**	**105317.5**

+ Duration (in milliseconds) taken to perform MOM defuzzification on a single object with multimodal data.

Crisp defuzzification output = {complete:1-3}, {unsure: 3-6}, {complete: 8-9}

4.5.2. Result

The MSA system collected dataset collected over 13,000 sensor events under an hour over two days. The evaluation results of the fine-grained AR algorithm are presented in Table 4.6. The results indicate the average accuracy of 83.33% and 100% to recognise F1 and F2 actions under six different *TIs*. The average duration of 24647.56 and 105317.5 milliseconds is recorded for F1 and F2 actions. Hence, indicating that there is a strong correlation with the increase in the number of rules in the set and time taken to perform MOM defuzzification. Moreover, the first defuzzification results at each run for F2 action are up to five times higher than reasoning time for the latter five *TIs*.

4.6. Summary and Future work

A fine-grained AR approach is presented in this paper to handles imprecise sensor information and the fusion of multimodal sensors on a single object to achieve higher accuracy. This paper makes three key main contributions towards AR at the multi-granularity level.

Firstly, a modelling approach for ADLs at coarse-grained and fine-grained action level is proposed. This modelling approach consists of developing OWL and fuzzy OWL model. The OWL model conceptualises ADLs at the coarse-grained level. This OWL modal consists of capturing context attributes (i.e., location, time interval and key objects), sensing environment, and the semantical relationships between everyday objects, sensors, and ADL. The fuzzy OWL model is used to define fine-grained actions using fuzzy set theory. The fuzzy set theory enables imprecise sensor data to be linguistically described within a gradual threshold using membership functions. Also, fuzzy rules are defined that fuses multimodal sensors attached to everyday objects (i.e., liquid level, temperature, accelerometer, and gyroscope on a kettle) to increase the accuracy of fine-grained action detection.

Secondly, a fine-grained AR algorithm is developed that utilises incremental pellet reasoning for reasoning with OWL model and fuzzyDL reasoner to perform defuzzification with fuzzy OWL and incoming non-/binary sensor events. The evaluation results indicate the average accuracy of 83.33% and 100% and an average duration of 24647.56 and 105317.5 milliseconds to perform multimodal sensor defuzzification for two fine-grained actions with 30 and 153 set of fuzzy rules.

Finally, a microservices-based system architecture (MSA) system was developed on two machines with real sensing environment consisting of non-invasive ambient sensors and embedded sensors. The MSA system successfully collected over 13,000 sensor events from 6 everyday objects with at least 19 individual sensors under an hour over two days. The future work will involve automating fuzzy rule developing process, optimising the accuracy and performance of the fine-grained AR algorithm for real-time system and compare against other DD approaches.

CHAPTER 5. PROBABILISTIC REASONING FOR UNCERTAINTIES IN HUMAN ACTIVITY RECOGNITION

Several issues such as sensor malfunction, dead battery, human errors, communication faults and environmental effects can create a lot of uncertainties leading to the concern about the reliability of the sensor data received or missing when performing HAR. To address this problem, this chapter proposes a probabilistic ontological knowledge modelling and reasoning approach to managing uncertainties in SH environment. It first analyses existing uncertainty theories and approaches developed in the past for AAL systems. Next, probabilistic ontology-based reasoning is proposed to model four abductive AR-related uncertainty attributes and user-feedback based knowledge learning mechanism. Multi-entity Bayesian Network (MEBN) theory is used as the core probabilistic knowledge modelling to capture four types of uncertainties (human errors, object functionality, SH devices and environmental based issues) and situation-specific reasoning. The approach leverages PR-OWL tool developed as a plugin for a popular artificial intelligence knowledge modelling tool, UnBBayes. A proof-of-concept case study is presented to model and reason with uncertainties within SH at fine-grained action level is illustrated. A discussion of advantage, open challenges, limitations and directions of future work is presented in this chapter.

5.1. Introduction to Uncertainties in HAR

An abundant of studies in the past have proposed diverse single-user HAR approaches that either focuses on tackling imprecise sensor measurements or uncertainties [175], [194], [195]. Both imprecise and uncertainties concepts are confused with being the same in the literature. However, they have some key differences. The impreciseness or vagueness occur when interpreting non-binary sensor measurements to be a member of a state to a certain degree, i.e., the cup is "*half*", "*nearly full*" or "*full*". In contrary, the uncertainties factors occur due to the results of unknown events that could happen in the future which cannot be measured or difficult to estimate, i.e., probability of communication network failing or sensor failure, and user forgetting to conduct actions. Consequently, both impreciseness and uncertainty factors are prevalent in a real-world environment, and it must be supported by the HAR algorithms to accurately estimate the activity occurring and provide accurate support[196]. Therefore, this chapter focuses on analysing state-of-the-art studies tackling uncertainties in the context of HAR and builds on approaches developed in CHAPTER 4 to recognise fine-grained user actions with impreciseness non-binary observations from the sensors.

SH environment is prone to be affected by several factors which can cause uncertainty with the data received and the confidence of AR results [169], [173]. Some of the factors creating uncertainty are mainly due to environmental, technological (i.e., sensor failure, low battery, interferences and packet loss), object (i.e. breakdown due to wear and tear) and human factors (i.e., mishaps/spillages, and forgetfulness). Figure 5.1 depicts these four types of uncertainty factors affecting the accuracy and reliabilities of HAR results.

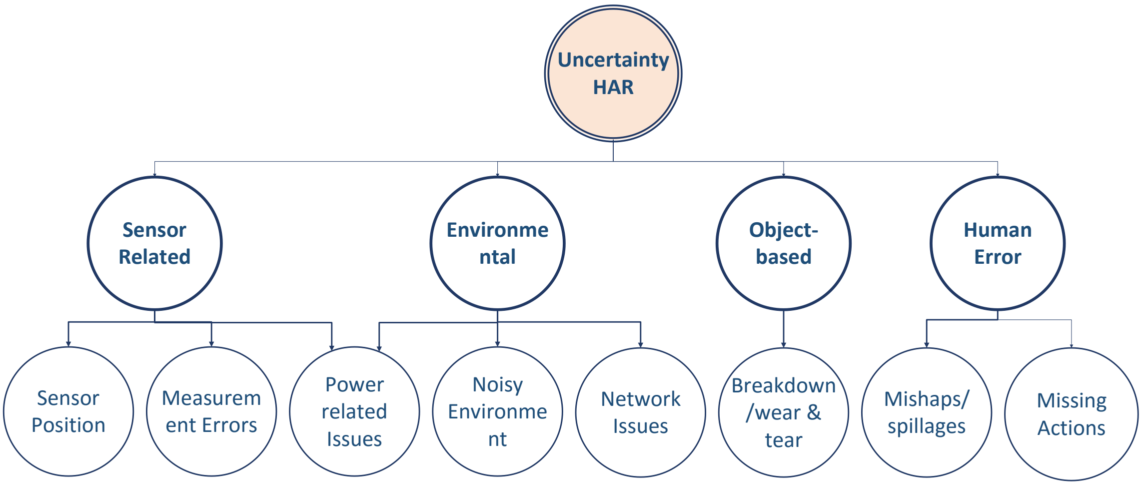


Figure 5.1. Typical uncertainty factors related to HAR within SH environment

Past studies have mainly dealt with uncertainty in HAR using deductive, inductive or abductive reasoning approaches. In the deductive reasoning approach, a mindful path from a *"general law to a specific case*" is followed[197]. For instance, the kettle is used to heat the water, the cup has hot water inside, so hot water must be from the kettle. The limitation of the deductive method is that it is assumed that the initial hypothesis is correct to examine the possibilities and reach a logical conclusion.

Conversely, the inductive reasoning approach assesses the situation from a "*specific case or a collection of observations to general law"*, i.e. from facts to theory[197]. For example, a fingerprint sensor has very low false detection error rate (fact 1), Bob has his thumb fingerprint enrolled in the sensor's database (fact 2), and therefore, Bob's thumb fingerprint will be detected by the fingerprint sensor (theory). The limitation of inductive reasons is that it allows the incorrect conclusion to be made even if the facts are true. However, this is approach is widely used to test the hypothesis in scientific research.

The abductive reasoning differs from deductive and inductive, where a real-world incomplete set of observations lead to a probable explanation for a "*propositions and their generalisation in a theoretical frame*"[197]. This case can be explained with observations that a given sensor battery is 50% after using it for two days and the manufacturer suggest that battery usually last up to 10 days. Hence, a generalisation can be made that a new battery is required and the sensor may be faulty or not energy efficient. Abductive reasoning is further investigation in this chapter, as it seeks explanation that best describes a state of events instead of the just matching pieces of evidence with a set of predefined laws/rules. Moreover, abductive reasoning is leveraged in this chapter to detect and explain uncertainties in missing, delayed or inaccurate sensor data in particular that can be used as a feedback to optimise the accuracy of HAR[57].

In the following sections, analyses of recent studies in modelling and reasoning with uncertainties in HAR are discussed in section 5.2. Based on knowledge gaps identified in the literature, a novel approach is developed in section 5.3. A case study to illustrate the applicability of the approach is illustrated in sections 5.4. A summary of this chapter is presented in section 5.5 with the discussions on limitations of the proposed approach and on future work.

5.2. Related Work

Allen Temporal Logic (ATL) has been effective in detecting missing actions in a mixed activities scenario using thirteen rules and time-series analysis[173], [198]. However, the shortfall of the ATL is the ability to explain the cause of the missing events, anticipate or predict actions that may have been conducted but not registered by the sensors. Therefore, ATL is ideal for detecting missing actions but not suitable for modelling and reasoning with uncertainty factors such as network delays or sensor errors in dynamic SH environment. Recent studies have dealt with such uncertainties in HAR by extending the capabilities of the ontologies and/or tightly integrating it with probabilistic theory[127], [128], evidential theory [129], [130], and fuzzy reasoning[57], [131]. The following sections will analyse the studies that adopted these theories.

5.2.1. Probabilistic Theory

The important work in probabilistic reasoning with ontology are BayesOWL[199], OntoBayes[200], Turambar[174], [201] and probabilistic OWL (PR-OWL) 2[202]–[208]. BayesOWL[199] applies a set of rules to OWL classes and generate two types of nodes, concept and binary relationships (L-nodes). The prior and conditional probabilities are given to the nodes. The limitation of BayesOWL[199] is that it can only define uncertainties to determine class membership of an individual[174]. OntoBayes[200] address the issue by focusing on relationships (object and data properties) and support multi-valued random variables but fails to model the relationship between classes.

Similarly, Turambar [174] presented an extension of SPARQL-DL and a reasoner to process probabilistic assertions of class and data/object properties assertions. However, Turambar suffers from handling dynamically changing situation and evidence collected from the SH as the queries are performed on a fixed number of nodes. Therefore, PR-OWL 2[202]–[208] has recently been introduced, which creates a situation-specific Bayesian Network (BN). A BN is a directed acyclic graph (DAG) with nodes containing a set of random variables, and a set of states to represent mutually exclusive and exhaustive possible values for some hypnotises[209]. PR-OWL 2 is based on Multi-Entity Bayesian Network (MEBN) that is supported by first-order-logic(FOL) and Bayesian probability theory[210]–[212]. Furthermore, PR-OWL 2 addressed the forward/backward compatibility issues between OWL syntax and SPARQL querying previously noticed in predecessor implementation[213]. Moreover, PR-OWL 2 offers an open-source, UMP-ST plugin [204] based on UnBBayes [214] to model uncertainties along with integrated Protégé (ontology editor). PR-OWL 2 has also been adopted by other domains such as maritime [212] and fraud detection in Brazil [206].

Several studies have applied probabilistic reasoning with the combination of ontology models and DD methods such as Markov Logic Networks (MLN) to describe uncertainties when recognising activities [37], [173], [194], [215]. For instance, work in [194] presents a probabilistic approach to segment continuous sensor events by leveraging ontological model and MLN to define activities, description logic (DL) rules for actions and associated uncertainty weights in the MLN. A Maximum-A-Posterior (MAP) query is performed over MLN to predict the most probable activity conducted by the user. This probabilistic approach was evaluated using WSU CASAS smart home dataset, and other DD approaches. The result indicated the proposed approach to achieve higher F-measure then Artificial Neural Network (ANN), Support Vector Machine (SVM), Bayesian Network (BN) and Hidden Markov Model (HMM). Moreover, ANN and SVM did not support uncertainty and incomplete data, whereas HMM performed better than BN in terms of handling uncertainties and temporal modelling. In addition, proposed MLN and ontology approach was compared with probabilistic ontology (PR-OWL). PR-OWL 2 is based on BN and could not match with weight learning algorithm of MLN that refines the ADL model with new weights.

In general, a probabilistic theory is adapted to represent uncertain outcomes based on objectively identifying patterns or frequencies from past events or subjectively based on expert knowledge to define the degree of belief. The objective probabilistic approach to define uncertainties in ADLs require real-time monitoring and analysing large amount of data produced by SH environments. Unfortunately, each SH environment is unique in terms of the size of the dwelling, types of sensors used, the total size of the sensing network and communications protocols adopted. Therefore, the challenge will be to develop a real-time objective probabilistic approach for each SH environment that is likely to shrink/grow dynamically over-time. This objective probabilistic reasoning is essential for critical systems requiring abductive reasoning approaches such as care homes and hospitals. Alternatively, the subjective probabilistic approach can prove to be beneficial for dwelling requiring necessary ambient and some embedded objects SH monitoring.

5.2.2. Evidential Theory

Dempster-Shafer Theory (DST) is also referred to as evidence theory developed to model and reason with uncertainty. DST was initially introduced by Arthur P. Dempster and later developed by Glenn Shafer as a framework to model uncertainty[130]. In DST, lack of information or missing sensor information is denoted as total ignorance with a weight (or belief), and accumulation of the weight with other pieces of evidence within a series of mathematical functions is calculated. The strength of DS theory is to handle conflicting sensor data [216] or sensor data fusion [130] problem by combining the pieces of evidence and arriving at a degree of belief to help the AR process.

Work in [130] developed a framework to model uncertainties at a low sensor level using DST and equally weighted sum operator(EWSO). The modelling process consists of developing an evidential network and mapping the belief values and actions for a set of ADLs. These belief values are then propagated by weighted sum operator to estimate the likelihood of activity occurring. Likewise, work in [129] presented a combination operator selection approach (COSA) to classifying uncertainty in an ontology tree and sensor data fusion. COSA incorporates DST, EWSO or maximisation operation (MO) in uncertainties modelling and reasoning process. Firstly, ontology and DST are used to model user actions in a given activity with mass function (belief values) between [0, 1]. Secondly, EWSO is a mathematical function to propagate uncertain concept with the piece of evidence/sensor states (frame of discernment) collected from compulsory set objects. Thirdly, MO is concerned with selecting most likelihood of activity occurring from the alternative activity. Subsequently, other studies have also explored defining uncertainties rules for activities such as belief rule-based inference methodology (RIMER) [217] and Weighted Average Combination Rule (WACR) [218]. These studies in common showed the usefulness of handling uncertainties in SH environment, however, binary sensors are mainly investigated, and the belief values are subjective to domain experts.

5.2.3. Fuzzy Theory

As discussed in the previous section 4.3.1.2, two types of Fuzzy set; Type-1 and Type-2. Type-1 Fuzzy set linguistically describe a vague concept to be a member of a given state to a certain degree using membership functions. Whereas, Type-2 fuzzy set introduces secondary membership functions where upper and lower membership boundaries are created when it is difficult to recognise simple fuzzy membership value for fuzzy terms/concepts. The region between the upper and lower membership boundaries is called the footprint of uncertainty[172]. The fuzzy ontology tool developed in [192] allows Type-1 and Type-2 fuzzy sets membership boundaries to be defined using `Datatype` annotations and `minValue/maxValue` attributes. Nevertheless, complementary FuzzyDL reasoner (adopted in CHAPTER 4) is currently unable to support reasoning with Type-2 fuzzy membership values.

Another work in [219] presented a fuzzy neural network (FNN) to recognise activities using voice speech and video for lip reading in an uncertain, noisy environment. FNN approach show improvement over just audio-based AR approach and applicability in a real-world setting. The limiting factor of this approach is that non-speech-based activity conducted in a noisy environment will not be recognised unless visual data is to understand other ADLs instead of lip-reading. Alternative work in [196] proposed knowledge modelling techniques using uncertainty ontology based on fuzzy Bayesian networks (UOFBN). UOFBN combine fuzzy ontology to handle imprecise nature of non-binary data and fuzzy Bayesian networks (FBN) to cope with probabilistic knowledge. The probabilities defined in FBN is used to create a conditional probability table (CPT) and Expectation-Maximization (EM) algorithm to estimate maximum-likelihood of an event occurring given the set of uncertain/incomplete data. The merit of UOFBN approach is shown by the simultaneous support to uncertainty and fuzzy knowledge modelling, however, practical application of the approach is yet to be realised with a tool/ontology editor plugin or feasibility in real-time AAL system.

In summary, ATL or such state-based techniques can answer “what” action or sensor data is missing from a given ADL but cannot explain “why” and “how” questions of uncertainties in events. Consequently, popular uncertainty theories (probability, evidential and fuzzy theory) and ontology modelling approaches were analysed as illustrated in Figure 5.2. The probabilistic theory is incorporated within the ontology modelling process (i.e., Turambar) or complementary with DD (i.e., BN and HMM) and KD approaches such as PR-OWL, OntoBayes and BayesOWL. Whereas, the evidential theory is driven by DST and rules where belief values are defined based on conditions of sensor states. Alternatively, the fuzzy theory is often used to express impreciseness of non-binary data/concept as gradian value to describe the weights of the uncertain action/activity in AR.

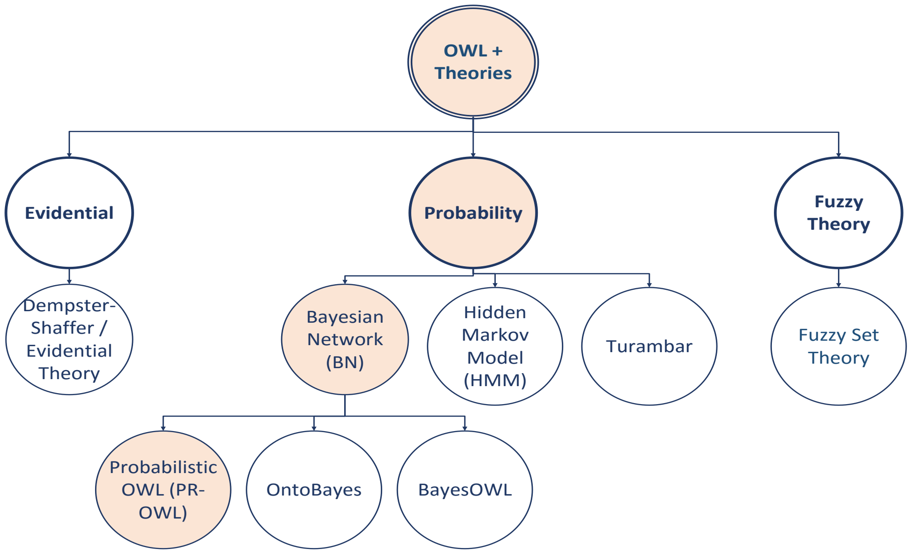


Figure 5.2. Overview of uncertainties theories applied to knowledge-based HAR modelling and reasoning

In general, these studies reviewed above mainly focus on inductive or deductive reasoning where many facts or general laws are already predefined for a given uncertainty factor. Although, the DD methods such as BN and MLN can support frequency or pattern detection to create abductive reasoning. Therefore, this chapter proposes to extend the probabilistic reasoning by combining creating a hybrid approach that adapts PR-OWL to predefine uncertainty laws/facts and seek user-feedback based to enrich uncertainty knowledge model in HAR.

5.3. Probabilistic Ontology based Uncertainty Reasoning

A probabilistic reasoning approach is proposed to extend the factual and fuzzy knowledge models developed in previous chapters to segment and perform AR at the fine-grained action level. For this, PR-OWL 2 is leveraged to complement crisp OWL and Fuzzy OWL model developed in section 3.3.1 and 4.3.1, respectively.

Figure 5.3 describes the uncertainty reasoning process where two external processes are providing inputs from AR results and SH raw data monitoring (at the bottom of the figure). Firstly, AR results from unfolding activities are analysed by the uncertainty reasoner to perform inductive reasoning with the known uncertainties within a given environment. The uncertainty reasoner creates/updates SSBN diagram and propagates the network based on four pre-defined uncertainty factors for each unfolding ADL. Section 5.3.1 provides details on modelling four uncertainty factors and propagating SSBN for each ADL. Secondly, SH devices and raw data output are monitored by the uncertainty reasoner to not only update the probability values in the model but also to identify the potential cause of missing sensor/user action. Section 5.3.2 presents details of four key parameters taken into consideration to perform abductive reasoning. These four key parameters are based on ADL time interval, missing sensor/action, environmental conditions and object functionality. Next, the recognised patterns based on abductive attributes and pieces of evidence collected are provided to the user to give useful feedback. The user feedback management and the process of updating the probabilistic model are discussed in section 5.3.3. Subsequently, details of the algorithm for the proposed approach is presented in section 5.3.4.

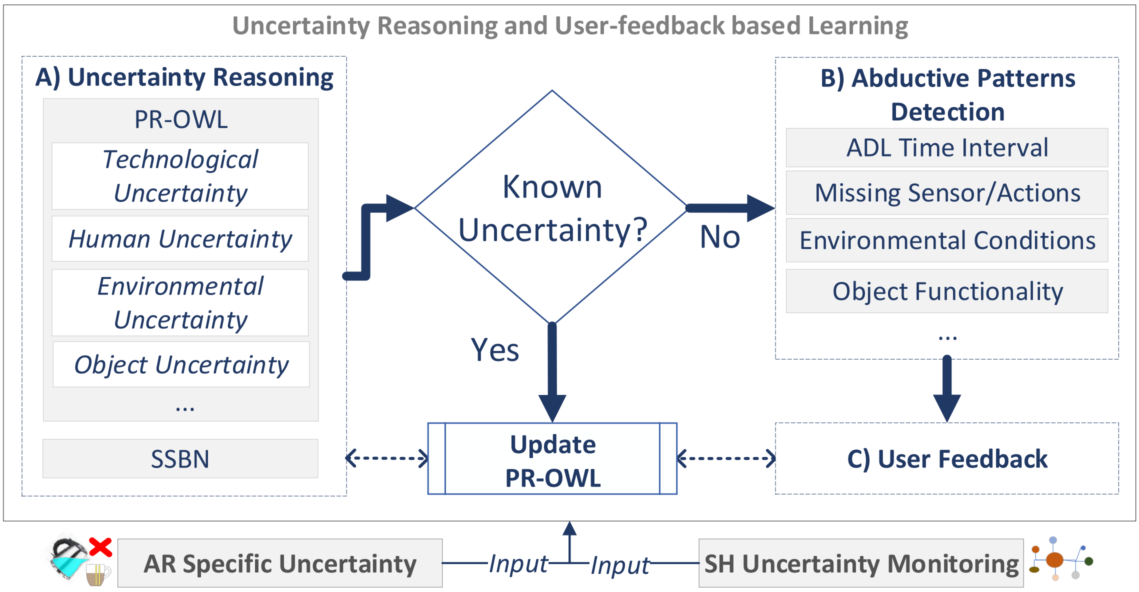


Figure 5.3. PR-OWL and User-feedback based Abductive Uncertainty Reasoning approach within SH and HAR context.

5.3.1. Smart Environment Uncertainty Factors Modelling and Reasoning

The process of developing PR-OWL model is to define an uncertainty variable as priori probability with MEBN fragments (MFrags) and complex MFrags groups to create MEBN theory (MTheory). The joint probability distribution of MTheory and MFrags allow creating situation-specific Bayesian network (SSBN) for each activity. Therefore, upon receiving a piece of evidence from SH, SSBN can be created, and probabilistic queries can be performed to determine the likelihood of an event/activity occurring. For instance, the goal is to determine if a sensor (S1) is sending faulty reading based on S1's performance attributes. The priori probability of S1's attributes can be defined in MFrags: battery life can be monitored, duration of sensor active, number of wireless sensors on the same frequency, prone to damage due to human consumption, manufacture sensor error rate. The evidence for S1's attributes can be added to SSBN and joint probability can be calculated to determine if S1 is faulty.

The common uncertainties caused in HAR are by the use of the everyday object ($objFactors$), human factors ($humanFactors$), technology-based ($techFactors$) and environmental factors ($envFactors$) are described using probabilistic theory, see equation 5-1. The $objFactors$ are those that can hinder the functionality of an object, i.e., due to wear and tear ($wearAndTears$) and manufacture defects ($defect$). The $humanFactors$ considered when conducting ADLs are accidents such as spillages of content or dropping the object with the content mid-action and missing out key actions. The evidence of spillage or drop is detected using IMU sensors when an object goes into freefall mode. The spillage or drops can occur in individuals with tremor, weak grips due to clumsiness or conditions (i.e., such as arthritis, tendinitis, and repetitive stress injuries). Another $humanFactors$ is the individual suffering from memory loss may forget to perform key actions based on the severity of their memory functions. This medical information about an individual can inform the knowledge engineer to predefine belief values and personalised the system. Furthermore, despite strategically positioning the sensor ($sPos_h$) on the object (Obj_c), individual may hold the object in incorrect orientation or outside the reading range of the sensor (i.e., capacitive touch or fingerprint sensor). Similarly, several $techFactors$ create uncertainties and reliability/trust issues with the data received from the noisy sensor network with the varying communication protocol. The wireless sensors often operate using batteries which can be consumed depending on the frequency of use and may provide false reading with a low battery level. Likewise, the $envFactors$ such as fire, flood, room temperature and humidity can have a severe impact on the operating conditions of the sensing devices.

$$\begin{aligned}\phi = \{\ & objFactors[defects, wearAndTears, \dots], \\ & humanFactors[Obj_c[sPos_h], action[damage, accidents, \dots], \dots], \\ & techFactors[faultySensor, lowBattery, networkDelay, noise, \dots], \\ & envFactors[\,fire, flood, temp, humidity, \dots], \\ & \dots \}\end{aligned} \qquad \textbf{5-1}$$

5.3.1.1. Probabilistic Ontology Modelling

To model these uncertainties, PR-OWL 2 is leveraged to captures these four types of factors in MEBN. These four types factors described in Table 4.5 are $humanFactors$ (A), $objFactors$ (B), $techFactors$ (C) and $envFactors$ (D).

PR-OWL 2 extends BN with FOL to create a MEBN logic for model complex knowledge. MEBN defines probabilistic knowledge as a set of MFrags to develop a minimum of one MTheory. MFrag contains four types of random variable (RV) nodes: resident, input, ordinary variable and context. MFrag containing RVs and their belief tables with probabilities make up the MTheory. The resident node is a yellow rounded rectangle node that consists of

RVs to form the core subject of the MFrag. The arcs pointing to resident nodes create conditional arcs and signify probabilistic dependence. The input node is a grey trapezoidal-shaped used for building relationships between the resident node from multiple MFrags. The input node can only point to other RVs but not to itself or from other RVs. Similarly, ordinary variable and context node are in green colour with a pentagonal shape. The ordinary variable node contains a variable or an instance of a class described in the ontology. The context holds Boolean RVs representing conditions (defined in first-order-logic (FOL) formulae) which must be fulfilled for the distributions defined in the MFrag to be valid. A context node cannot have any arcs pointing to or from it.

Table 5.1. Overview of Uncertainty Types Considered when Modelling ADL Knowledge Base.

Type	Uncertainties Description	Evidential theory
A	Accidents: Spillage /Drop	Object free fall detection
A	Missing key actions	Mandatory/optional events and identify dependencies using Allen's temporal rules
B	Utility device breakdown: caused by wear and tear, leading to incomplete actions.	Device functional status, main power supply status (if applicable), under warranty (durability), fragility in use level
C	Sensor failures: manufacturing defect, battery low, maintenance, out of range	Maintenance report: last battery change, estimated power consumption
D	Undesirable operating conditions causing sensor damage/failure	Water-related activities, the brute force required, incorrect temperature
D	Frequency noise, network	Number of radio-frequency devices, the magnetic field
D	Uncontrolled events: power cuts, storms, earthquakes	Power cut, storm and earthquake statistics in the area.

***Note**: Uncertainty types: Human error (A), Object-based (B), Sensor-based (C). Environmental (D)*

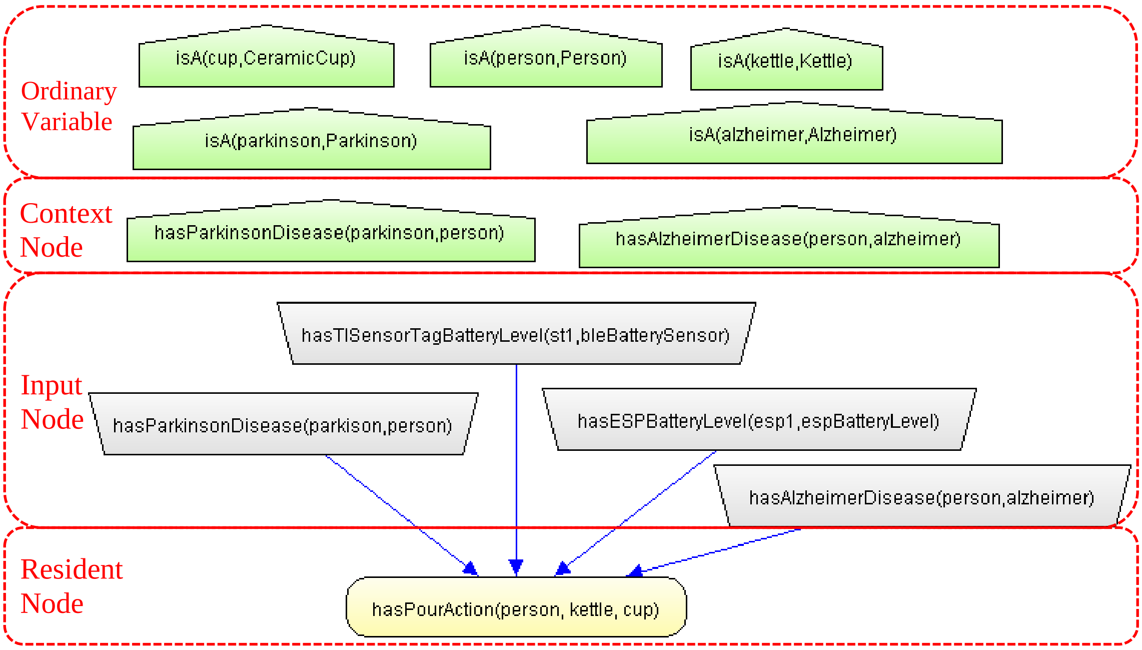


Figure 5.4. An example of *MakeTeaADL* MFrag comprising of uncertainties when detecting *Kettle* pouring action to *CeramicCup*.

Figure 5.4 presents an example of *MakeTeaADL* MFrag consisting of four types of RVs to define human factors and technology-related uncertainties when estimating kettle pouring action to the cup. The ordinary variables or instance of the OWL class are initially created, which can be used as a parameter by the context and resident nodes. In this case, instances of *Kettle*, *Person*, *CeramicCup*, *Parkinson* and *Alzheimer* classes are added to MFrag. The context nodes are object or data properties (*hasParkinsonDisease* and *hasAlzheimerDisease*) defined in another *HumanFactor MFrags* resident nodes and linked to *MakeTeaADL* MFrag as context nodes. Similarly, input nodes in *MakeTeaADL* MFrag are TI Sensor Tag and ESP microcontroller battery level are linked with Technology MFrags. The *hasPourAction* resident node has arcs from the four input nodes and three ordinary variables as parameters. The local probabilistic distribution values for *hasPourAction* resident node can be defined with nested if-else conditions of four input nodes as defined in Figure 5.5. This nested if-else condition checks if the BLE TI SensorTag and ESP microcontroller's battery levels at the first level using *bleBatterySensor* and *espBatteryLevel* ordinary variables. Based on the state of the two sensor's battery level variables, i.e., if the battery value is low or empty, second-level nest conditions are executed which checks if the person has Alzheimer or Parkinson diseases. In essence, this nested if-else condition gives a higher probability for the user pouring action *successfully* if there are no known diseases and the sensor battery levels are not low or empty.

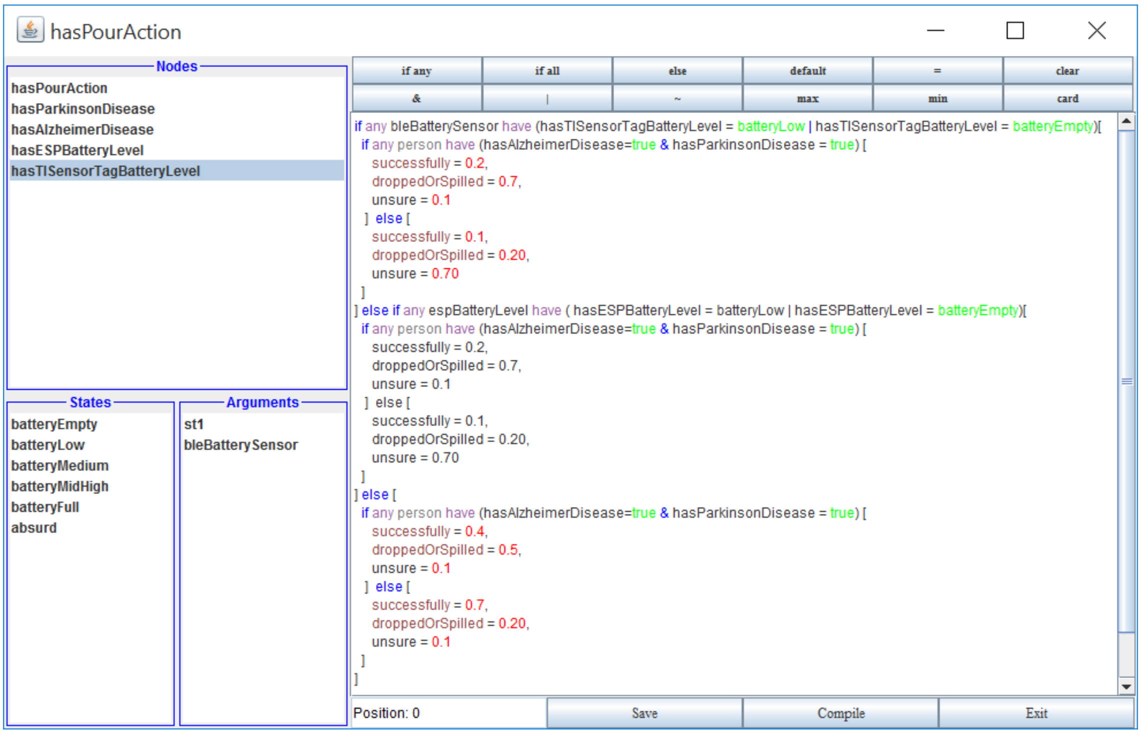


Figure 5.5. Editing *hasPourAction* probability table based on the known disease of the user and sensor battery levels.

Consequently, MFrags can be created for other four uncertainty factors and ADL of interest. The probabilistic distribution of four types of uncertainty is currently pre-defined. However, with more data over time and online/offline activity learning algorithm, these probabilities can be dynamically updated. In this approach, we propose attributes for abductive reasoning and user feedback mechanism to update the probability distribution table (more details in section 5.3.2). The evidence collated from the SH environment is assessed for uncertainties and provided to the SSBN to create/update belief tables for a given ADL dynamically. The input RV in BN requires a crisp input of an imprecise sensor data (i.e., battery level low, medium and high). Hence, fuzzyDL modelling and reasoning results from fine-grained action level AR (discussed in CHAPTER 4) are used before providing the input to the SSBN model. The effects are propagated with other belief tables in the BN. The propagation results will show the overall estimation of the ADL occurring based on known uncertainties.

5.3.1.2. Progressive Propagating Evidences

As the sensor observations are received, data are filtered and combined with fuzzy reasoning, an SSBN is dynamically created and updated with more evidence for each ADL. The effects are then propagated with all the belief tables in BN in order to calculate the probability of the ADL occurring. There are two types of propagation methods, diagnostic (backward) and prognostic (forward) reasoning. The diagnostic reasoning is traditionally used for decision-making to identify the root cause of the failure based on symptoms or pieces of evidence collected from the SH environment. Therefore, the diagnostic approach is used to enter pieces of evidence collected based on how user’s actions from the SH and AR results to calculate the overall effect in recognising ADLs. In contrast, the prognostic reasoning is concerned with entering evidence about the causes and predicting the likelihood of the future outcome. For example, if the sensor battery is low, the next set of sensor data may be unreliable due to a higher probability of error in data measurements and loss in data packets during transmission. Consequently, the prognostic approach is used to perform offline or online monitoring of not only how technology is responding but also the other factors that enable us to answer or predict the future of a given event occurring. Therefore, the prognostic reasoning process is responsible for updating the probabilistic distribution table for resident nodes in the MEBN knowledge model and diagnostic reasoning to add evidence to SSBN created dynamically.

5.3.2. Abductive Pattern Recognitions

The goal of the abductive pattern recognition is to collect pieces of evidence of unexplained events and develop a self-learning algorithm or ask for user feedback. Although the development of self-learning algorithms is out of the scope of this chapter, we propose a user feedback mechanism based on four types of abductive attributes. These four types of attributes considered are based on ADL time interval, object functionality, missing sensor/action, and environmental conditions. A summary of these four types of attributes is defined in Table 5.2.

The time interval attribute within which a set of ADLs are commonly performed is grouped in order to identify uncommon activities performed by the user. The change in user preferences, a sequence of conducting ADLs and even specific actions with an alternative object within ADL overtime is common. As these changes are not be explicitly defined in the ADL knowledge model, the system will not be able to reason with data received from SH environment. Therefore, requesting the user to provide feedback on the unknown actions or sensor events at ADL and action level. For example, the use of cinnamon, ginger and peppercorns while making tea in the morning may be a result of change user preference and these actions are not part of other ADLs at generally occur before lunch or dinner.

The everyday object functionality attribute is concerned with factors such as wear and tear of the objects over time that undermine its designed operations. The lifespan of non-/electrical appliances varies amongst manufacturers and subject to nature or frequency of use. Consequently, following the correct procedure to use an object may not achieve results. For instance, a user purchased the kettle three years ago and used it to boil the water, but on a given day, the kettle heating coil failed to heat the water, or there was an unscheduled power cut when making tea.

Table 5.2. Four Types of Abductive Attributes Identified to Recognising of Uncertainties and Requesting User Feedback.

Abductive Attributes	**Techniques**	**Grouping/Patterns**	**ADL Levels**
ADL time interval	Time series	Creating four periods (i.e., morning, afternoon, evening and night) during the day where a set of ADLs are typically conducted and identifying the ADL performed outside this period.	Activity and Action level
Object functionality	Frequency	Monitoring functional properties of the everyday objects, i.e., if the user has turned the kettle on and it has heated the water at the correct temperature.	Action level
Missing sensor/action	Frequency	Tally network size, measure network speed and reliability of packet delivery to verify the functionality of the sensor. Hence, enabling to determining user forgetting to conduct actions.	Action level
Environmental condition	Sensing Attributes	Monitor changes in air quality, temperature, humidity and light and identifying a median, minimum, and maximum values during the day.	Activity and Action level

The missing sensor or actions attribute inspects SH network by performing active diagnostics on individual sensors to assess speed, packet delivery error rates and compare ground truth of the sensor measurements with a similar or more reliable sensor on the network. These diagnostics tasks enable analysing the functional properties of the SH network and determine if the user has forgotten to conduct the actions. Over time, a record of the forgotten set of actions for ADLs will be reviewed by the user to verify the condition of the SH devices and suggest reasons behind unregistered or forgotten actions.

The environmental conditions such as air quality, temperature, humidity and light within a room can provide new insights into why certain actions were missed or not recorded. For instance, 7am heating system timer makes the kitchen room temperature to be very hot which may influence the user to have cold coffee, drink orange juice or have yoghurt more regularly before leaving the house in winter. This activity pattern can explain why the user is at risk to develop any illness over time. Furthermore, regular exposure to high temperature and humidity in the room can impact the operations of the sensing devices. Therefore, abductive reasoning will identify the low and high and peak values of such environmental conductions and seek feedback from the user on their health and other reasons for their actions such as lactose intolerance.

5.3.3. User Feedback Management

Based on the four abductive attributes and patterns identified in section 5.3.2, a user is requested to provide details reasons at activity and action levels with the probabilities of such patterns reoccurring in the future. If the reasons provided are known, the probabilistic distribution tables in MFrag will be updated accordingly. Otherwise, new MFrags for the missing activities/actions are created with four types of RVs and probability distribution table defined, respectively. The information provided by the user is assumed to be factual and correct. However, this assumption is too strong and require additional measures to verify the knowledge and ensure the knowledge model to reusable on other users.

5.3.4. Algorithm with Uncertainty Reasoning

The algorithm for PR-OWL based uncertainty reasoning and abductive reasoning based on user feedback is presented in Table 5.3 as pseudocode. The algorithm requires four inputs: AR results (`arResult`) conducted in sections 3.3.3 and 4.3.4, SH data (`shData`), candidate ADL (`adlClass`) of interest, and user details (`user`). The output (`prowlResult`) of the algorithm is produced for storage and future analysis. The algorithm is divided into four parts: finding (1) missing actions, (2) analysing uncertainty reasoning with missing actions, (3) performing abductive reasoning and (4) updating feedback from the user.

In the first part, lines 1-2, list of missing actions from AR results (`arResult`) are retrieved using *getMissingAction* function. The `getMissingAction` takes list of observed actions (`arResult.getActions()`) and `adlClass` of interest and stores it to list (`mActions`).

In the second part, lines 3-11, each missing action (`act`) is analysed based on four uncertainties factors modelled in section 5.3.1. On line 4, temporary Boolean variables to indicate four types of known uncertainty factors related to the action and instance of `PROWLData` class (`prowlResult`) is created. Next, line 5, iterate over each *act* from `mActions`, and line 6 retrieve sensor attached to a given object used for the `act`. The sensor data (*s*) extracted is used to check for technical factors and human factors on lines 7-8. The functions, `checkTechnicalFactors` and `checkHumanFactors` identify if the factors related to the sensor or object is known and create/update SSBN accordingly. A Boolean result indicates if the factors are known in the PR-OWL model or not so that abductive reasoning can be conducted (in the third part). Likewise, lines 9-10, perform object (`checkObjectFactors`) and environmental (`checkEnviroFactors`) factors check based on missing action using object details and `shData`.

Table 5.3. Pseudocode for Handling Uncertainties with Probabilistic Ontology Reasoning in ADL Processing Thread (PT_x)

ALGORITHM: ***Input***: arResult, shData, adlClass, user
Output: prowlResult

```
1  //1) Find missing actions in arResult
2  List mActions = getMissingAction(arResult.getActions(), adlClass);
3  //2) If missing action found, find sensor attached to object user should have interacted with.
4  Boolean a, b, c, d = false; PROWLData prowlResult = new PROWLData();
5  for Action act: mActions
6      Sensor s = getSensorDetails (act.getEverydayObject());
7      a = checkTechnicalFactors(s);
8      b = checkHumanFactors(act.getEverydayObject(), s, user);
9      c = checkObjectFactors(act.getEverydayObject(), shData);
10     d = checkEnviroFactors(act.getEverydayObject(), shData); endif
11     //3) If uncertainty factors are OK but action is still missing, conduct abductive reasoning
12     if !(a, b, c, d)
13         List tsa = runADLTimeSeriesAnalysis(act, s, adlClass, shData);
14         List obja = runObjectFunctionalityAnalysis(act, s, adlClass, shData);
15         Performance p = runSensorQualityAnalysis(s, shData);
16         Boolean eia = runEnvironmentalImpactAnalysis(s, tdbGetLocation(s)); endif
17     //4) If abductive explanation identified, ask for user feedback.
18     if tsa.isEmpty() && obja.isEmpty() && p==null && !eia
19         prowlResult.addMissingAction(act);
20     else
21         updateProbabilisticTable(prowlResult, getUserFeedback(tsa, obja, p, eia)); endif
22 endfor
23 return prowlResult;
```

In the third part, lines 11-16, if any of the four uncertainty factors are unknown, abductive reasoning based on four attributes is conducted to collect potential pieces of evidence that can lead to new findings with the help of user feedback in the fourth part. In lines 13-14, time series analysis and object functionality testing are performed using `runADLTimeSeriesAnalysis` and `runObjectFunctionalityAnalysis` functions. Both functions take in the *act*, *s*, *adlClass*, and `shData` parameters and return a list of evidence to suggest a possible cause.

Similarly, on line 15-16, diagnostic functions for sensor performance quality (*runSensorQualityAnalysis*) and environmental impacts (*runEnvironmentalImpactAnalysis*) are conducted and stored in respective variables. The location of the sensor is retrieved from the metadata stored in the triplestore using *tdbGetLocation* function.

The final part, line 17-21, check for any abductive reasoning identified for the missing action and if found, user feedback (*getUserFeedback*) sought. Otherwise, *prowlResult* is appended with missing actions. The user feedback is handled by the *updateProbabilisticTable* function on line 21 to update PR-OWL model with suggested probabilistic distribution table in existing MFrag or create a new one. This part 2-4 of the algorithm is repeated for each missing action and the result is stored/broadcasted to the user on line 22-23.

5.4. Use Case Study

To illustrate the applicability of modelling uncertainties using probabilistic reasoning, four factors affecting detection of the kettle "*pouring*" action into the cup while making tea is developed using PR-OWL ontology. For this, PR-OWL plugin in UnBBayes software application is used to create MTheory consisting five MFrags as shown in Figure 5.6. These MFrags comprising RVs and probabilistic distribution tables of nodes are stored as a sperate extension "*.ubf*" from "*.owl*" file. Additionally, observations collected from the SH devices are stored in "*.plm*" extension file consisting instances of sensor and object classes with their states.

The first MFrag, *Technology_MF*, consists of four ordinary variables for ESP8266 microcontroller and TI sensor tag instances with their battery level status. The two resident nodes, *hasESPBatteryLevel* and *hasTISensorTagBatteryLevel*, consisting of respective instances of four ordinary variables to identify the individual object and hold their current battery status. These two resident nodes will be later used by other MFrags as input nodes to provide information and conditions in which probabilistic distribution table can be defined. In this case, *MakeTea_MF* MFrag uses two resident nodes in *Technology_MF* MFrag as input node to influence defining probabilistic values for determining the likelihood of *pourAction* occurring.

Similarly, other three MFrags, *HumanError_MF*, *Environmental_MF*, and *EverydayObject_MF*, are created with RVs to define uncertainties with a respective object in use. The fifth MFrag, *MakeTea_MF*, links all the four MFrag by defining inputs nodes containing residents' nodes with probability values from the four MFrags. Therefore, enabling the *pourAction* context node in *MakeTea_MF* to define probabilistic dependent values based on conditions of the four factors as described in section 5.3.1.1 and depicted in Figure 5.5. An SSBN is created by running a query on any of the context node defined in the MTheory with pieces of evidence collected and stored as instances in the knowledge base (i.e., by creating new instances or loading "`.plm`" file).

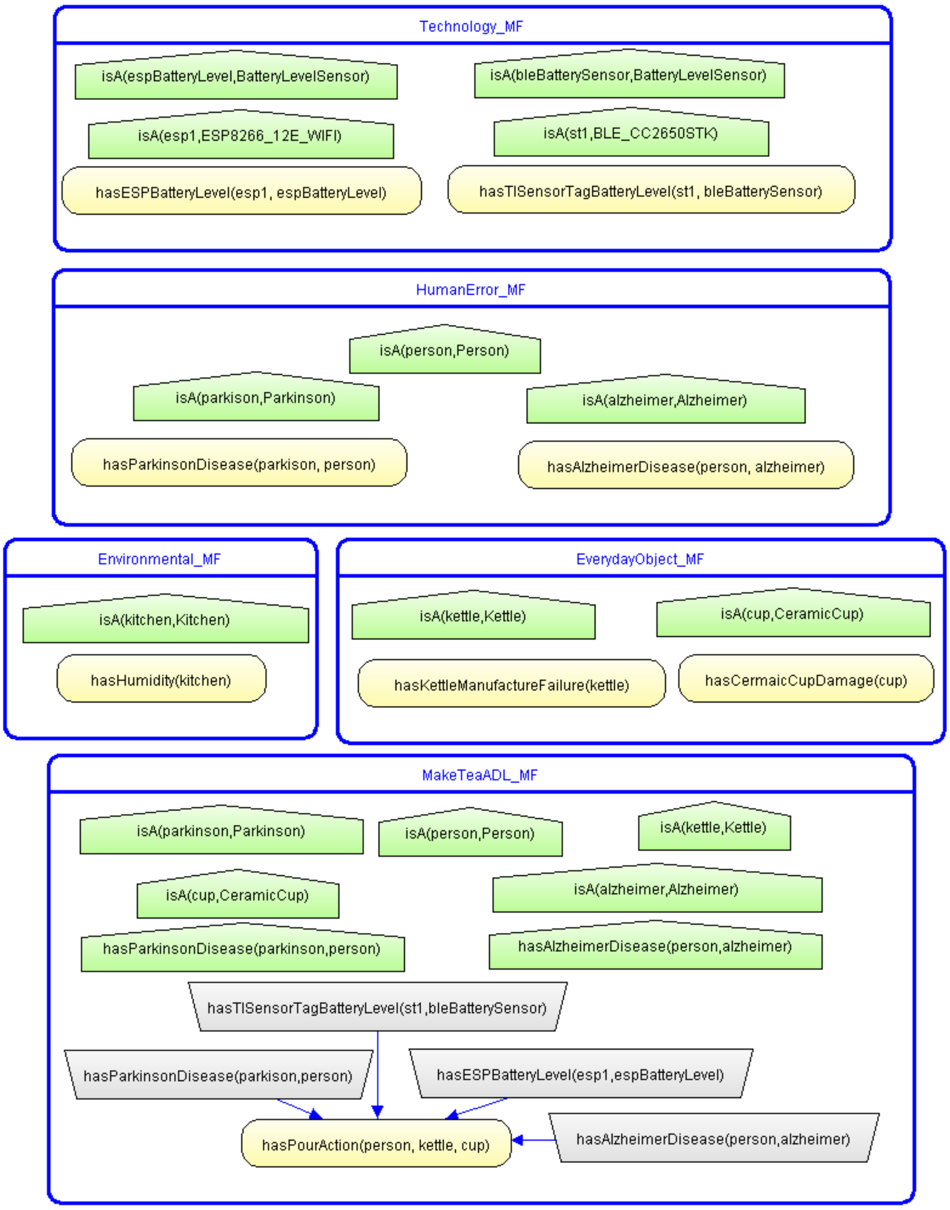


Figure 5.6. An Example of MTheory Containing Four Uncertainty Factors to Detect Kettle Pouring Action into the Cup When Making Tea.

5.4.1. Discussions

The main benefit of the proposed probabilistic reasoning and user feedback mechanism is that existing crisp, and fuzzy knowledge is extended with the ability to formally conceptualise uncertainty factors. However, the inheriting limitation of manually effort and performance is still the main obstacle of ontological based uncertainty reasoning. Potential avenues to combine

KD and DD techniques[57] to evolve the initial model can provide opportunities to learn and evaluate the finite knowledge base. Some of the open challenges that require further investigations are validating new knowledge, assessing trust/credibility of the source, handling conflicting concepts/facts defined in the knowledge, finding semantic duplications, and managing/tracking change in the ontological model.

5.5. Summary and Future work

To summarise, existing OWL and Fuzzy Ontology-based approaches lacked considering taking uncertainties factors influencing the estimation of AR results. Hence, to extend the expressivity of the ontological model and incorporate the uncertainty factor in the AR process, PR-OWL based on multi-entity Bayesian network (MEBN) is proposed in this chapter. Four types of uncertainties factors were considered within PR-OWL: human error (A), object functionality (B), technology (C) and environmental (D). One of the benefits of using MEBN is that subject-specific Bayesian network (SSBN) is dynamically created and updated as the evidence from the sensor are observed. The information from the AR process defined in CHAPTER 3 and CHAPTER 4 is used by the probabilistic reasoning algorithm to not only identify missing actions but also interpret non-binary sensor data. The affected belief tables for the BN network can be propagated to calculate the overall probability of a given ADL occurring. For a proof-of-concept, a PR-OWL ontology is developed as a case study to assess the likelihood of the user with Alzheimer and Parkinson disease to pour hot water from the kettle to cup given the fact that battery levels of sensors measuring the user interaction are also low.

In addition, to peace evidence together from the missing actions or predict future potential problems in AR, four abductive attributes are identified. These four abductive attributes are based on (1) change in ADL pattern using time intervals, (2) object functionality, (3) missing sensor/action from a technology perspective, and (4) environmental conditions. To perform abductive attributes-based reasoning, previous SH data and live diagnostics on responsiveness and reliability of the sensor network are proposed to be conducted over-time. Based on the findings of abductive reasoning, the user-feedback mechanism is proposed to enable the user to meta-data about the anomaly in ADL pattern and the frequent missing of actions or unknown actions conducted in a specific time interval.

Finally, the proposed approach requires further advancement by integrating open source UnBBayes and PR-OWL APIs into the real-time system to analyse the performance and feasibility of the approach in comparison to other state-of-the-art approaches. Furthermore, open challenges in tracking and evolving finite set of knowledge will be explored using DD and KD approaches using semantic models.

CHAPTER 6. FRAMEWORK FOR SINGLE-USER ACTIVITY RECOGNITION WITH FUZZY AND UNCERTAINTIES KNOWLEDGE

Current studies have mainly focused on developing accurate HAR algorithms with factual knowledge using KD or DD approaches. However, limited studies integrate both, imprecise measurements of multimodal sensors and uncertainty factors when recognising HAR in an AAL system. Hence, this chapter investigates and develops a framework that leverages the KD approach to describe unambiguous information with Web Ontology Language (OWL), imprecise knowledge with fuzzy OWL and uncertainty with probabilistic OWL. The key components of the framework are organised within a microservice system architecture (MSA) to improve performance, availability, and maintainability over-time. A single user AR algorithm is proposed based on the proposed ontological-based modelling and reasoning framework. This framework is applied to a kitchen-based application scenario for a single user AR and provide evaluations on preliminary findings.

6.1. Introduction

The real-world smart environment is filled with ambiguous sensor data and uncertainties that impact all aspects of Human Activity Recognition (HAR) tasks in the context of Ambient Assisted Living (AAL) system. The heterogeneous sensing environment output non-binary measurements subjected to human interpretations. Hence, creating a challenge for the activity recognition (AR) algorithms to reason with this non-binary information to infer single user activity or fusing multiple sensing attributes for higher accuracy. CHAPTER 4 analysed existing studies to handle non-binary information and proposed a fuzzy ontological (Fuzzy OWL) modelling and reasoning approach for fusing multimodal sensor data for higher accuracy in detecting user actions at a fine-grained action level. The Fuzzy OWL based modelling and reasoning approach provided promising result to achieve express imprecise knowledge and achieve higher accuracy in recognising user activities at coarse and fine-grained action levels. However, the main shortfall of CHAPTER 4 is the ability to support uncertainties of events that may or may not occur due to several factors such as failure in sensing/transmitting data on time, low battery levels of wireless sensors, damage to object in use or forgetfulness due to human chronic illness such as Alzheimer. Therefore, these uncertainty factors pose essential questions on the reliability and trust in the information gathered from the smart environment and AR results.

Consequently, CHAPTER 5 reviewed studies adapting state-of-the-art uncertainty theories such as probabilistic theory, evidential theory and fuzzy theory. As a result, probabilistic theory based probabilistic ontology (PR-OWL) modelling and reasoning approach was proposed. In addition, four types of uncertainties factors commonly present in the smart environment, HAR and AAL system, in general, were identified and used for modelling a probabilistic reasoning purpose. These four types of uncertainty factors considered are technological, human, object functionality, and environmental. PR-OWL is based on Multi-Entity Bayesian Network (MEBN) which creates a network of nodes to implicitly define a joint probability distribution over possibly infinite numbers of hypothesis or uncertainties. PR-OWL enable Situation-specific Bayesian Network (SSBN) to be created based on the pieces of evidence collected from the smart environment and propagate the effected nodes. PR-OWL approach was evaluated and showed the applicability of developing the uncertainty model with four types of factors presented in a given smart environment and recognising Activities of Daily Living (ADL) at activity and action level.

However, limitations of both of the approaches is that modelling, and the reasoning process is focused on either impreciseness or uncertainty factors. Moreover, limited tools for modelling and reasoning with fuzzy and probabilistic knowledge are available that is compatible with each other and easy to integrate within the AR process. Hence, more investigation is required to bridge these two types of knowledge within the AR process. Consequently, this chapter analyses state-of-the-art studies tackling impreciseness and/or uncertainties in section 6.2. Based on the findings, a novel ontological framework is developed in section 6.3. The evaluation of the proposed framework and discussions are provided in section 6.4. This chapter finally presents a summary of the contribution and future research direction in section 6.5.

6.2. Related work

Recent studies have highlighted three main challenges faced when developing HAR algorithms to analyse the sensor data: (a) modelling complex relationship between SH devices, ADLs and user; (b) handling ambiguous data and fusing multiple sensor data; (c) handling uncertainties.

6.2.1. Reusing and Defining Semantical Relationships between Entities

Firstly, an SH environment is composed of heterogeneous sensing and communication technologies developed by individual manufactures with their own IoT-enabled solutions. This can not only create interoperability/scalability/reusability challenges with cross manufacturer devices but also modelling and reasoning with the incoming sensor data. As discussed in section 2.3, a wide range of off-the-shelf and commercial SH devices are available in the market which requires third-party APIs integration and compatible aggregators to support various communication protocols. The relationship between the entities, SH devices, and ADL knowledge need to be coherently described to enable reasoning algorithm to deduce in/explicit links and infer ongoing activities. Therefore, work in [220] presents a framework to semantically describe the SH environment and domain-specific knowledge, ADLs, to perform AR based on SPARQL queries and clustering. This approach requires manual effort to develop the ontological model from scratch. In addition, several conflicts and duplications in knowledge can occur when conceptualising such domain-specific, hence, making it difficult to reuse or share with others. Fortunately, Semantic Sensor Networks (SSNs)[221] vocabularies have been developed by SSN Incubator Group and W3C. SSN provides a comprehensive set of classes and relationships (object and data properties) to describe the system (properties, features and conditions), deployment environment and sensor details (including observations/actuation values, sampling procedures, and results). Moreover, SSN ontology has a subset ontology containing core classes and properties called SOSA (Sensor, Observation, Sample, and Actuator) which lightweight and self-contained. Thus, SOSA ontology help kickstart the projects by focusing on describing knowledge of interest and integrate full SSN ontology as the system and project mature over time.

6.2.2. Non-/binary Data Fusion

Secondly, a single object of interest can have two or more types of sensors outputting non-binary interaction information which is subjected to interpretation, challenging to fuse the data and make multicriteria base decisions. The initial challenge is to handle the imprecise/vague measurements collected from non-binary sensors are subjective in nature[222]. As discussed in sections 4.2 and 5.2.3, fuzzy set theory has widely applied to handle impreciseness within the context AAL system [166] and other domains such as flight booking[172], and diabetic mellitus[171]. The subsequent challenge is the fusion of multimodal sensor data and multicriteria decision analysis (MCDA)[223], [224]. For which, fuzzy theory has also been applied in [183]–[185]. However, the critical limitations for adopting fuzzy set theory with ontological models is the availability of modelling and reasoning tools to develop a fuzzy knowledge base. Recent efforts made by Umberto and his team to develop a fuzzy ontology plugin for popular ontology editor Protégé [186], and fuzzyDL[187] reasoner. The details of the tool can be viewed in [188]. To the best of our knowledge, these tools have not been used within the context of describing fuzzy sensor measurements and fusioning multimodal sensor data to achieve object-usage level HAR.

6.2.3. Uncertainty Frameworks in AR

The third challenge is to incorporate uncertainties with the sensor data caused by environmental, technological and human factors[173]. As discussed in the previous CHAPTER 5, several DD and KD studies have been explored to describe uncertainties when recognising activities [37], [173], [215]. However, the studies undertaken to resolve all three challenges have been investigated in insolation and fall short in distinguishing and/or recognising the need for all three elements within an AAL system.

A hybrid activity recognition framework (ARF) is presented in [37], [173] to semantically learn activity models. The objective of the ARF is to address temporarily and hierarchically related semantic queries under uncertainty by adapting the probabilistic event and dynamic relationship learning methods. Therefore, avoiding the need to manually model the ADLs, temporal and hierarchical dependencies. The approach relies on propositional formulas and the weightings definition to describe uncertainties of user actions. To evolve the formals, pattern learning techniques are adapted. Moreover, inspiration from thirteen Allen's temporal rules, a semantic model was developed to conduct reasoning with MLN method. Likewise, work in [215], presented ontology-based AR with the MLN method for probabilistic reasoning to handle uncertainty and refining the inferred activity (by instances consistency checking). Nevertheless, ARF and MLN based approaches provide little support to process imprecise data using fuzzy knowledge.

Work in [213] presents a framework dubbed, ByNowLife, which integrates OWL and Bayesian Network(BN) for simultaneous logical and probabilistic reasoning. The application of the framework was demonstrated using two case studies: banking stocks investment problem and social customer relationship management in a context of higher education and university. However, in the context of AR, one of the key limitations of the BN based approach is that it fails to detect missing sensor events. To address this issue, work in [169], proposed action sequence based missing sensor event detection using HMM and Dempster-Shafer theories to combine evidence. However, prescribing or estimating all the possible sets of combinations of a given activity is unrealistic to model due to its exponential growth and cause performance and scalability issues as pointed out in [37], [173]. Therefore, this chapter proposes to separate mandatory and optional actions for a given activity to detect missing sensors using Allen's rules. Furthermore, the evidence collected from the sensor-based multi-modal smart environment will be separated, filtered, analysed by fuzzy set theory and the results will be passed to situation-specific BN (SSBN) so that the propagation results on all the uncertainties can be calculated on a given ADL.

6.3. A Framework for Semantic-enabled HAR in AAL system

A semantic-enabled AAL framework is proposed to recognise single-user ADLs within a real-time smart environment. The framework leverages KD approach for modelling and reasoning with crisp, imprecise, and uncertain sensor data. A holistic view of the framework is presented in Figure 6.1. The framework is organised in five key components using microservices-based system architecture (MSA). The SmartWeb application programming interface (API) is a web service which fulfils all the requests made by external client devices from the system. The SmartWeb API liaises with four primary internal web services to route the client's requests to relevant web service(s). These four internal web services are application, service, sensing platform and data storage. The role of application API is to manage user profile details, take actions to prompt or alert users for any anomalies and provide AR results to the users. The core function of the service API is to analyse user activities with the data collected by the sensing platform API and user profile from the data storage API. The responsibilities of each internal web services are further discussed in CHAPTER 8. However, this chapter focuses specifically on service API to bring together crisp (σ), imprecise (π) and uncertainty (ϕ) knowledge based to perform single-user AR with unobtrusive multimodal sensing method in the following sections.

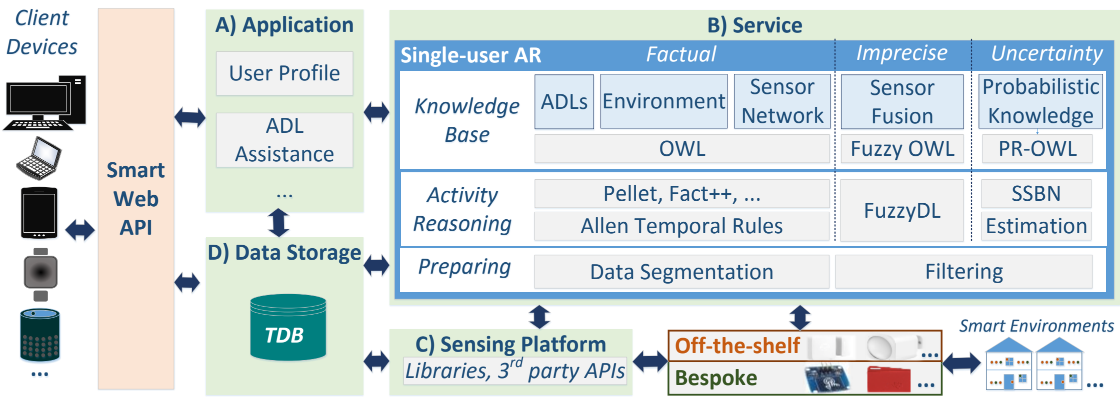


Figure 6.1. Framework for Semantic-enabled Imprecise and Uncertainty Knowledge in SH Environment and AAL System

The service API is the core module of the AAL system. The ADL assistance feature in the application API relies on the service API to analyse the sensing data received from the sensing platform API. The service API contains three main components required for HAR: preparing (pre-processing) of the sensor data, knowledge base and activity reasoning engine.

The preparing component consists of filtering noisy sensor data and segmenting sensor events into the respective set of ongoing ADLs for further data analytics. The sensor data collected from sensing platform API such as accelerometer and gyroscope are prone to drift in their reading over time. Hence, filtering and smoothing techniques such as complementary and Kalman filter are required before performing activity recognition algorithms. The filtered observation values for a set of segmented sensors for a given ADL are evaluated using sliding windowing process; more details in section 4.3.4. The semantical segmentation [153] approach, proposed in CHAPTER 3, is responsible for separating and group sensor observation based on object/entity's relationship with ADL descriptions specified in the knowledge-base ($\mathcal{KB}$). Both generic and user-specific preferences knowledge is utilised to segment each sensor observation into a relevant set of ADL queues using incremental pellet reasoner to perform terminology-box reasoning (T-box) and assertion-box (A-Box) reasoning.

6.3.1. Conceptualising Crisp, Imprecise and Uncertain Knowledge

The $\mathcal{KB}$ defined in CHAPTER 4 and CHAPTER 5 is extended to conceptualise σ, π and ϕ within an ontological modelling framework as denoted in equation 6-1.

$$\mathcal{KB} = \{\sigma, \pi, \phi\} \quad \textbf{6-1}$$

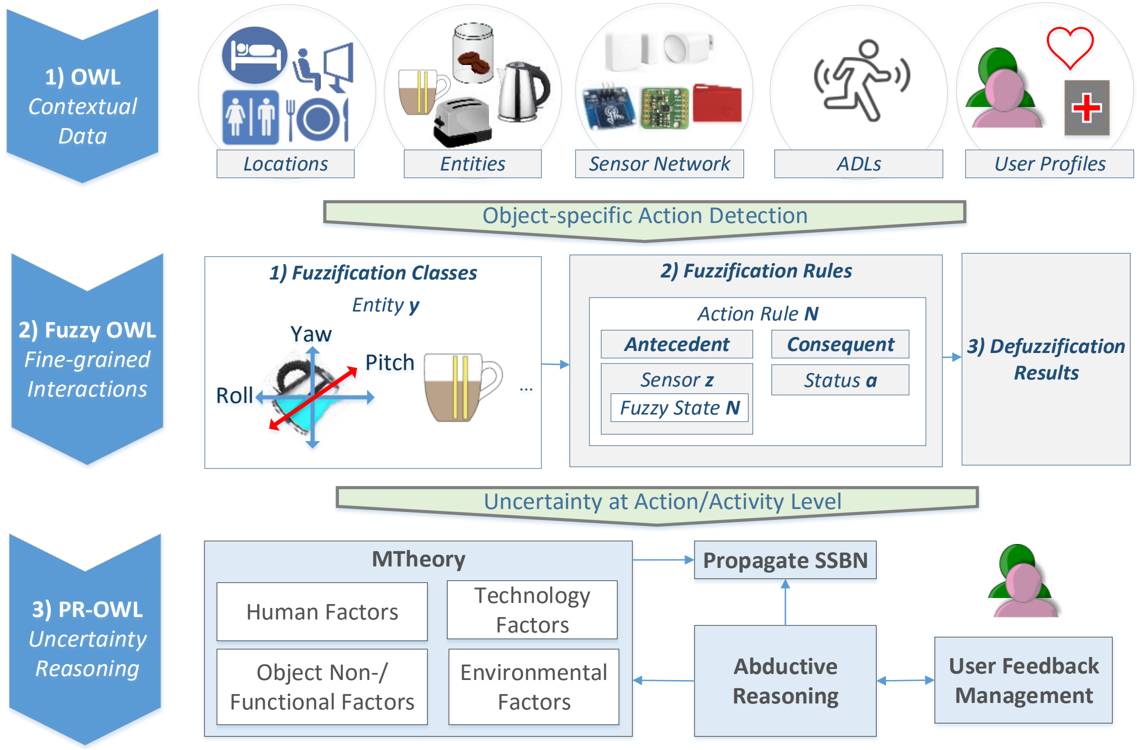


Figure 6.2. Ontological modelling framework to capture factual, imprecise concepts and uncertainties in the context of AR and AAL.

Figure 6.2 elaborate on the ontological modelling framework to develop σ, π, and ϕ concepts in $\mathcal{KB}$ with their relationships in three phases. In the first phase, contextual data of a given establishment where the system is deployed will be described at multiple levels of abstraction to form the backbone of the $\mathcal{KB}$. The contextual data consist of physical or non-physical entities in a given environment, ADLs being conducted in different locations with those entities, sensing technologies employed and user profiles to form a σ knowledge set.

$$\mathcal{KB} = \{\sigma, \pi, \phi\}$$

The second phase of the framework goes beyond assuming if an action with an entity has been conducted from the binary sensors and more towards recognising user interactions and actions with entities at atomic level (i.e., actions that cannot be further decomposed). For this, pieces of evidence from multimodal sensors are required to be collected, interpreted and reasoned to detect individual user actions. Hence, the subjective non-binary concepts are initially conceptualised and then fused with multiple pieces of evidence from the sensors to recognise object-specific user actions. Consequently, enriching the σ knowledge set with a set of π knowledge containing object-specific actions required to conduct ADLs. Section 6.3.1.1 further elaborates the relationship mapping between σ and π knowledge.

In the third phase, uncertainty factors affecting the reliability and accuracy of the AR results based on σ and π knowledge set is conceptualised. Each uncertainty factors can impact recognition results of one or more activities at atomic action levels during a given time instance. Therefore, uncertainty factors and their impacts in recognising ADLs need to be described at multiple levels. Section 6.3.1.2 provides details on the adapting probabilistic ontology modelling approach to define uncertainties and enrich the σ and π knowledge model.

Similarly, section 6.3.1.3 bind together σ, π and ϕ knowledge in a uniform modelling framework for single user AR at multi-granularity action level. For this, core concepts between the three types knowledge and their relationships are mapped and described at an abstract level. Furthermore, strengths and weaknesses of currently available tools that influence the modelling process are presented with a mapping solution.

6.3.1.1. Crisp and Imprecise Knowledge Modelling with SSN

The σ and π knowledge-based developed in CHAPTER 3 and CHAPTER 4 are integrated when recognising activities at multi-granularity levels. The σ knowledge model consist of crisp concepts and relationships between ADLs ($\mathcal{ADL}_i$), the environment ($\mathcal{Env}_a$), and sensors network ($\mathcal{SN}_d$) as described in section 4.3.1.2 and denoted in equation 4-2. In order to conceptualise diverse $\mathcal{SN}_d$ deployed in complex smart environment, SSN vocabulary is proposed to be integrated to increase the expressivity of the ontological model. SSN vocabulary enables describing the sensor network capabilities from individual sensor's sensing attributes to operating conditions under with a given platform is deployed. In general, the SSN vocabulary comprises of eight interlinked modules: *deployment, system, system property, condition, feature, procedure, observation/actuation/sampling*, and *result*. The classes and relationships (object/data properties) for each of these modules are comprehensively described in [225]. However, this section describes key classes and properties from the SSN vocabulary required to build the σ and π knowledge specific to AAL application is depicted in Figure 6.3.

In the top part of Figure 6.3, a snapshot of how SSN/SOSA vocabulary is used to describe ADLs, environment with location and objects information, and sensor network with aggregators, sensors/actuators and their observation properties. For illustrative purpose, the main classes from the lightweight SOSA vocabulary are used to describe sensing platform (*sosa:Platform*), inbuilt sensors (*sosa:Sensor*), types of feature attribute sensor captures (*sosa:FeatureOfInterest*) and its data (*sosa:Observations*/ *sosa:ObservaleProperty*).

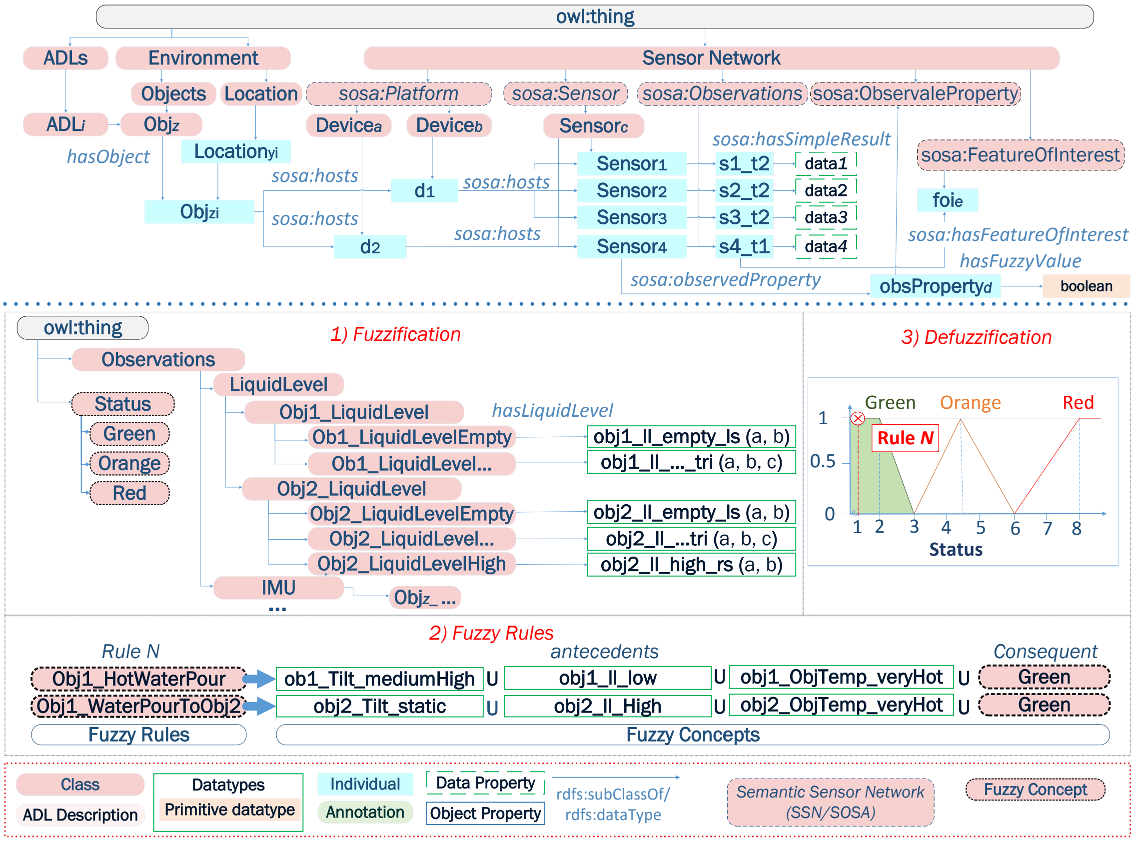


Figure 6.3. Integrating Semantic Sensor Network (SSN) Vocabulary within Crisp (top) and Fuzzy ontology (bottom) Modelling Processing

The *sosa:Platform* class enables devices which host sub-platforms, provides communication and sensing capabilities with other devices (i.e., smartphone) to be described. Each *sosa:Platform* device can host (*sosa:hosts*) more than one *sosa:Sensor*, *sosa:Actuator*, *sosa:Sampler* or sub *sosa:Platform* and can be deployed (*ssn:inDeployed*) in a given environment (*ssn:Deployment*). The *ssn:Deployment* class *ssn:inDeployed* property belongs to *deployment* module of SSN vocabulary used to describe uncertainties in a given environment where a system is configured (more details in section 6.3.1.2). Each *sosa:Sensor* can observe changing attributes/states of the object/entity with *sosa:ObservableProperty* class and *sosa:observes* object property relationship. The *sosa:ObservableProperty* class allow property or characteristics of an object (i.e., distance and temperature). The *sosa:FeatureOfInterest* class is a thing for which the changing attributes/state is being observed by *sosa:ObservableProperty* class in order to arrive at the result (*sosa:Result*), or whose attributes/state is manipulated by a *sosa:Actuator*. For example, when measuring the temperature of the kettle, the temperature is the *sosa:ObservableProperty*, 20 Celsius is the *sosa:Result* of *sosa:Observation*, and the kettle is the *sosa:FeatureOfInterest*. Likewise, a microcontroller class ($Device_{a\text{-}b}$) is a type of *sosa:Platform* where ESP8266 12E instances (d_1, d_2) equipped with sensors ($sensor_c$) such as distance and temperature sensors ($Sensor_{1\text{-}4}$) with respective observed property ($obsProperty_d$) with the results (s_1-s_2) collected between two-time instances (t_1, t_2). The result value of the sensor is stored using *sosa:hasSimpleResult* data property and primitive values (i.e., string, integer, Boolean). However, the user-specific data structure can also be created using *sosa:Result* class *sosa:hasResult* object property.

The imprecise nature of non-binary sensor data is indicated using *hasFuzzyValue* data property as a characteristic with the *sosa:observableProperty* instance of $obsProperty_d$. Each of these fuzzy characterises is defined using fuzzy OWL vocabularies and fuzzyDL plugin compatible with Protégé 4.1 to the π knowledge model as depicted at the bottom of Figure 6.3. The π knowledge model includes fuzzy observable properties to describe the non-binary sensor s_f state of the crisp $\mathcal{Env}_a$ or Obj_z classes, i.e., if the *room* temperature is "*cold"*, "*warm*" or *"hot"* and *kettle* object is *"empty"*, *"half-full"*, or *"full"*. Hence, the fuzzy set theory is applied to not only model fuzzy concepts but also fuse the states of multimodal sensor data to recognise activities at fine-grained action level.

As described in section 4.3.1.2, the fuzzy knowledge base is created in three steps: fuzzification, rules generation and defuzzification. In the fuzzification step, the observable states (i.e., *LiquidLevel*) of an Obj_1 (i.e., *Obj1_LiquidLevel*) consists of a set of fuzzy states, s_f, (i.e., *Obj1_LiquidLevelEmpty*, *Obj1_LiquidLevelLow, … Obj1_LiquidLevelHigh*). These fuzzy states are described using fuzzy membership functions (*d*) and modifiers (*mod*). The membership functions are *trapezoidal*, *triangular*, *left-shoulder*, *right-shoulder*, *crisp interval*, and *linear*. The modifiers are *linear* and *triangular*. Both *d* and *mod* take different shapes when defining gradual boundaries between states. The $triangular\,(x, a, b, c)$ and $trapezoidal\,(x, a, b, c, d)$ membership functions defined in equations 6-2 and 6-3 are used to define fuzzy concepts. The variable x in both functions is the input value and other letters for gradual boundaries. As illustrated in the fuzzification step in Figure 6.3, a new datatype is created to define each fuzzy membership functions for *LiquidLevel* fuzzy sensor states of an Obj_1, i.e., *Obj1_ll_empty_ls(a,b), Obj1_ll_low_tri(a,b,c),…, Obj1_ll_high_rs(a,b)*. Likewise, other Obj_z's liquid level, gyroscope and accelerometer states can be defined as fuzzy concepts.

$$triangular\ (x, a, b, c) = \begin{cases} 0, & x \leq a \\ (x-a)/(b-a), & a \leq x \leq b \\ (c-x)/(c-b), & b \leq x \leq c \\ 0, & x \geq c \end{cases} \quad \textbf{6-2}$$

$$trapezoidal\ (x, a, b, c, d) = \begin{cases} 0, & x \leq a \\ (x-a)/(b-a), & a \leq x \leq b \\ 1, & b \leq x \leq c \\ (d-x)/(d-c), & c \leq x \leq d \\ 0, & x \geq d \end{cases} \quad \textbf{6-3}$$

In the second step of fuzzy knowledge modelling, a set of $rules$ are created with one or more fuzzy membership states defined as the datatype for multiple sensing attributes of an object to define a condition under an action is complete. These $rules$ enable defining conditions in which a given fine-grained action (fa) is complete by fusing multiple sensing parameters attached to the object using operators such as union (∪). The fuzzy rules consist of IF (antecedent) and THEN (consequent) statements. For instance, *Obj1_HotWaterPour* rule defines condition of a fa. This condition for *Obj1_HotWaterPour* rule states that pouring hot water from kettle is only complete `IF kettle hasLiquidLevel obj1_ll_half-full ∪ hasObjTemperatureLevel obj1_objTemp_veryHot ∪ hasTiltPosition obj1_ll_mediumHigh THEN hasStatus Green`. The final step is defuzzification, where the rules and membership functions are used to identify if the given sensor inputs are associated to fa to a given degree between 0 and 1. The common defuzzification methods available are Centroid Of Area (COA), Bisector Of Area (BOA), Mean Of Maximum (MOM), Smallest Of Maximum (SOM) and Largest Of Maximum (LOM)[166].

6.3.1.2. Uncertainty Knowledge Mapping

The ontological modelling framework integrates uncertainties factors using probabilistic theory and probabilistic ontology (PR-OWL) modelling process. CHAPTER 5 provided details of the process of developing PR-OWL based on the σ and π concepts and a kitchen-based uncertainty example as a case study. However, this section focuses on bringing σ, π, and ϕ knowledge together to define uncertainty at activity and action level.

Figure 6.4 presents a conceptual view of activities defined in σ ontology, actions at fa ine-grained level in π and ϕ factors defined at both levels using OWL(c), Fuzzy OWL (fo) and PR-OWL (po/po2), respectively. The prefixes such as *c, fo, po/po2, ssn/sosa, owl, rdfs* are shorthand for full uniform resource identifier (URI) to distinguish individual class and properties belong to a given vocabulary or three types of $\mathcal{KB}$. As discussed in section 6.3.1.1, crisp concepts are defined in OWL and fine-grained level actions in Fuzzy OWL. The PR-OWL reuse the classes and properties created in the OWL (justification provided in Section 6.3.1.3) to develop a probabilistic model by creating MTheory with a set of MFrag. An MFrag contain a set of *po2:OrdinaryNode*, *po2:ContextNode* and *po2:InputNode* with arcs to describe uncertainty at activity (*c:ADL*$_k$/*fo:ADL*$_k$) and action (*fo:ADL*$_k$*_Act*$_m$ / *po:ADL*$_k$*_Act*$_m$) level.

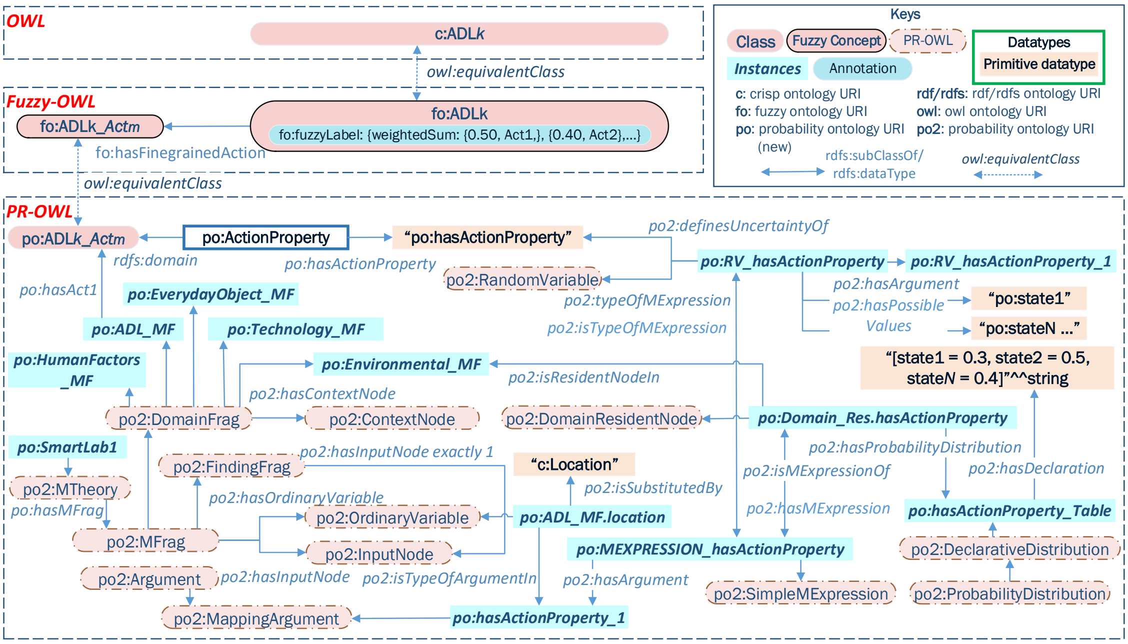


Figure 6.4. Defining Uncertainties using Probabilistic Ontology at Activity and Action level

In Figure 6.4, there are two types of *po2:MFrag*: *po2:DomainFrag* and *po:FindingFrag*. The *po2:DomainFrag* is used to store expert in the domain to model the uncertainties. The *po:FindingFrag* is used to enter evidence in MBEN MTheory and allow probabilistic algorithm to revaluate the SSBN based on the new evidence. Depending on number of pieces of evidence required for *po:FindingFrag*, *po2:inputNode* with argument and ordinary variable (*po2:MappingArgument*, *po2:OrginaryVariable*) instances of classes such as *c:Location* is created with a *po2:hasInputNode* object property with cardinality restriction of exactly 1. In the contrary, *po2:RandomVariable* is used by *po2:DomainFrag* as an argument (*po2:hasArgument*) to create *po2:ProbabilisticDistribution* table with possible values (i.e., *po:batteryLow*, *po:batteryMedium*, *po:batteryHigh*, …, *po:stateN*) with *po2:hasPossibleValues* data property relationship. These possible states and probabilities values add up to 1 and stored as a string using *po2:hasDeclaration* data property relationship. Subsequently, five types of *po2:DomainFrag* are created: four for activity level (*po:HumanFactors*, *po:EverydayObject_MF*, *po:Technology_MF* and *po:Environmental_MF*) and one for action-specific (*po:ADL_MF)* uncertainty factor. These set of *po2:DomainFrags* are interlinked by *po2:ContextNode* to create a probabilistic Bayesian network diagram which is used to create a situation-specific Bayesian network (SSBN) based on the sensor inputs.

6.3.1.3. Bridging the Gap between Multi-layered Ontological Knowledge

The tools used to develop $\mathcal{KB}$ is FuzzyDL plugin in Protégé for σ and π knowledge, and PR-OWL plugin in UnBBayes for ϕ knowledge, respectively. However, these tools are currently incompatibility with each other, underdevelopment and have limitations which causes σ, π, and ϕ knowledge to be modelled in isolation.

The main benefits of creating isolated ontology with domain-specific knowledge are that classes and relationships become lightweight, easy to reuse in other domain and provide more opportunities to track and maintain the model more efficiently. In contrary, one key limitation is that currently, the FuzzyDL plugin is only supported by older version of Protégé 4.3 and cannot import external vocabularies (i.e., SSN/SOSA) when annotating or converting ontology file to fuzzyDL syntax file and run fuzzyDL reasoning. In addition, fuzzyDL reasoner requires dependency software such as legacy version of Gurobi 6.52 to optimise in mathematical calculation when running defuzzification. However, PR-OWL plugin in UnBBayes supports importing external vocabularies and even has embedded Protégé environment and inbuilt ontology reasoners. Hence, enables fuzzy ontologies developed using Protégé and fuzzyDL plugin to be modified using UnBBayes and PR-OWL plugin. UnBBayes and Protégé are both developed using Java and they are open source, yet, neither of tools currently support modelling σ, π, and ϕ concepts in a single platform.

Consequently, a mechanism is developed to bridge the σ, π, and ϕ knowledge within the ADLs and SH to create a reusable ontological model for AR. The two key issues considered when creating the mapping between the three different ontologies are duplications and disjointed/fragmented knowledge. The duplication in knowledge is created when same object is syntactically described using different words but mean the same or the same class/property/instance is required in the individual ontological model, i.e., in π, and ϕ. Henceforth, to avoid duplications in knowledge between three ontological models, *owl:equivalentClass* object property is used with the full external URI. For instance, *c:ObjectN* class defined in crisp ontology is equivalent to *fo:ObjectN* and *po:ObjectN*.

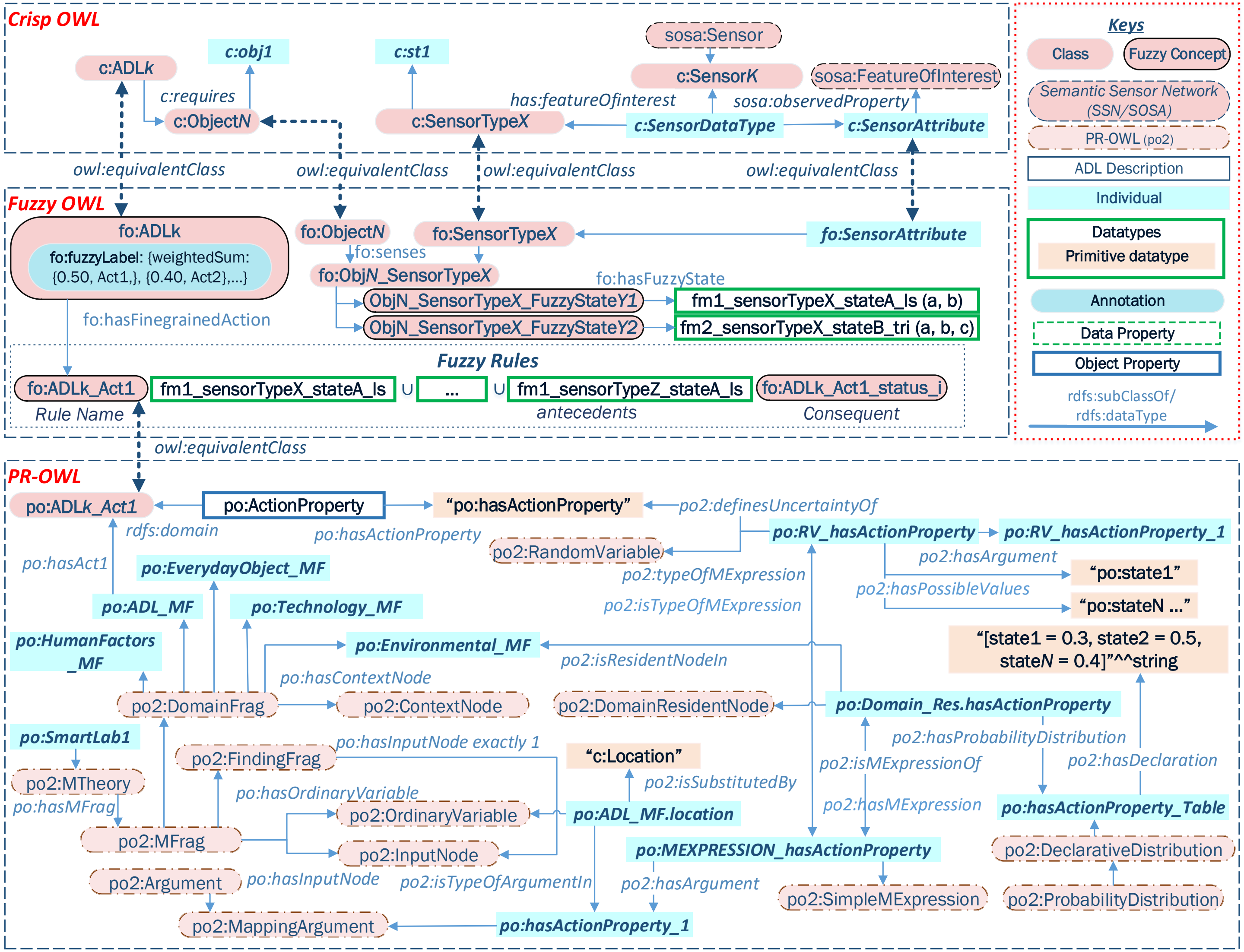


Figure 6.5. An Ontological Framework for Modelling Crisp (σ), Fuzzy (π), and Uncertainty (ϕ) Knowledge for HAR in an AAL System

The disjoint in knowledge modelling and reasoning process is created with excessive overlapping domain knowledge between multiple models and their inconsistency in intended use for that entity/properties. For instance, the *c:ADLk* class is defined to contain abstract description of the set of entities required to complete the activity compared to *fo:ADLk* which set of key actions required with their importance weighting and *po:ADLk* class with uncertainty factors. Hence, when developing the reasoning algorithm, correct class URI belonging to a specific type of ontology is used when querying or iterate through the knowledge model. Figure 6.5 describes the overall mechanism of the relationship mapping between the σ, π, and ϕ concepts in the ontological $\mathcal{KB}$ framework

6.3.2. Activity Recognition with Ontological Modelling Framework

The activity reasoning engine performs three main tasks to reason with σ, π, and ϕ knowledge. In the first task, the reasoning engine analyses each action within the activities at multi granularity level: coarse-grained action (ca) and fine-grained (fa) as discussed in CHAPTER 4. The ca activity level mainly considers attributes under which a given activity must be fulfilled. These attributes are location, time interval and key objects. The attributes for each activity are stored in the ontological model/graph-based database and can be queried using SPARQL Protocol and RDF query language (SPARQL) or description logic (DL) query approach, respectively. Furthermore, to detect missing actions or sensors for a given activity, aforementioned attributes are used to create a set of lists to describe mandatory/optional dependencies and 13 Allen rules to identify missing actions; more details in section 6.3.2.1. Secondly, the fa are analysed using π model to detect incomplete actions from the set of actions described in the model. The fuzzy rules are used to combine multiple sensor data at a given time window and infer if the fuzzy state corresponds to fa. In addition, a set of fa are defined with importance weighting to calculate overall completion of an ADL using *fo:fuzzyLabel* annotation and *weightedSum* concept type; further elaborated in section 6.3.2.2. Thirdly, the uncertainty related to the events is taken into consideration based on attributes identified from task one to perform activity level uncertainty reasoning and action-specific uncertainty reasoning based on the result of task two. Section 6.3.2.3 provides further details in performing task three.

6.3.2.1. Detecting Missing Sensors

The mandatory actions or context events are those that are considered essential in order to conduct the activity and they follow specific sequences. To describe sequence dependencies in OWL, *hasMandatoryDependences* object property, is used within the class description of the ADL. For example, while making tea ($A1$), the occupant must "*start*" by picking up the kettle

(A) to fill up with a tap on (B) and heat the water "*before*" turning the kettle switch (C) on. On the other hand, optional actions are those that allow the occupant to conduct other actions without conducting the optional action. To describe optional sequences dependencies in OWL, *hasOptionalDependences* object property, is used within the class description of the ADL. For instance, the occupant may "*start*" by opening the cupboard (D) to take the tea mug out (E) and spoon (F) out; here during F during D or D contains F can be used. However, occupants may also pick up the mugs and spoon from the mug and cutlery stand by the kitchen sink/platform. Hence, opening cupboard (D) action is defined as optional for $A1$.

Table 6.1. Example of Finding Missing Actions and Potential Sensor Failure.

$A1$ (partial OWL class description): = *hasMandatoryAction* some (A *or* B *or* C *or* E) *hasOptionalAction* some (D *or* F) *hasMandatoryDependences* some (($A\ start\ B, B\ before\ C$), …) * *hasOptionalDependences* some (($D\ start\ E, D\ contains\ F$), …)
Conducted action sequences: $ABCDEF$
Observed event sequences: $ABEF$
Following steps are taken to find missing actions/failed sensor: 1. Retrieve both types of dependencies for $A1$ activity 2. Check mandatory dependencies for A, B and E. o Mandatory action A start B OK. o Mandatory action B before E FAIL. o Mandatory action B before C EXPECTED. o Mandatory action B before E FAIL. 3. Check Optional dependencies for E and F o Optional action B before E FAIL, o Optional action E before F FAIL. o Optional action D contains E EXPECTED. o Optional action Dcontains F EXPECTED. 4. Check last message or request status from the sensors attached to C and D within the threshold amount? → YES = missing action, NO = sensor failure
OWL Example: *$hasMandatoryDependences\ some$ (((($starts\ some\ A$) $and\ before\ some\ B$)), $and\ finishes\ some\ C$))

In order to identify the missing sensors/actions, let us assume, both of the mandatory and optional example actions (A-F) are conducted sequentially and observed by the sensor network but failed to administer C and D actions. Hence, the observed $ABEF$ sequences of events are analysed by retrieving both types of dependencies for A1 activity. The mandatory actions A and B satisfy Allen's "*start*" rule but fail to match any actions sequences between B and E. The expected mandatory action is B before C. However, it fails to see any mandatory dependencies between B and E. The action E is not part of any mandatory dependency sequences, hence optional dependencies are check for the rest of the events, E and F. The optional dependencies sequences failed to match action link with B and E, E and F. However, both E and F actions expected action D to be conducted but missing from the observed sequences. The sensor event log can be further checked to see when the last message was received from the sensors attached to C and D objects when a given threshold or request status check in order to determine if the sensor active or failed. Table 4.6 describes the above kitchen scenario and the process of finding the missing sensors and potential sensor failures.

6.3.2.2. Detecting Incomplete Actions

The incomplete actions of one or more objects with multiple sensors observations are detected using fuzzy reasoning. The actions are observed and matched against a set of criteria at prior (or initial), present, and post states with fixed three-second window size. The set of criteria for prior state of action are used to detect changes at the end of the present state and the expected set of goals after having performed the action at post-state. For example, to detect "*pouring*" action, the prior state assumes that kettle is on the base and turned off, the kettle has some water (i.e., half the amount measured with picofarads (pF) value of 15), water temperature is very hot, and the cup is not full. As the user moves the kettle, the position (roll, yaw and pitch) is observed until the end of the present state time window and compare if the kettle has at least been moved and tilted in proportion to the half of the water amount (i.e., triangular function with (27.61, 20, 10) pF values). The fuzzy thresholds-based reasoning can help identify to what membership degree the movement of the kettle when half full fall under. The post-state of the action is to verify if the water level in the kettle has decreased, the cup has been filled (i.e., not empty) and the temperature was raised (i.e., *hot* or *very hot*). Likewise, other fine-grained actions such as "*drinking*" from the cup are detected with relevant sensors. Figure 6.6 describes "*pouring*" and "*drinking*" measurements at the initial and new state of the kettle and cup with sensors (i.e., liquid level, position data and object temperature) attached. In addition, a snapshot of fuzzy reasoning based on rules describing a set of criteria to be satisfied at three states is presented in Table 6.2.

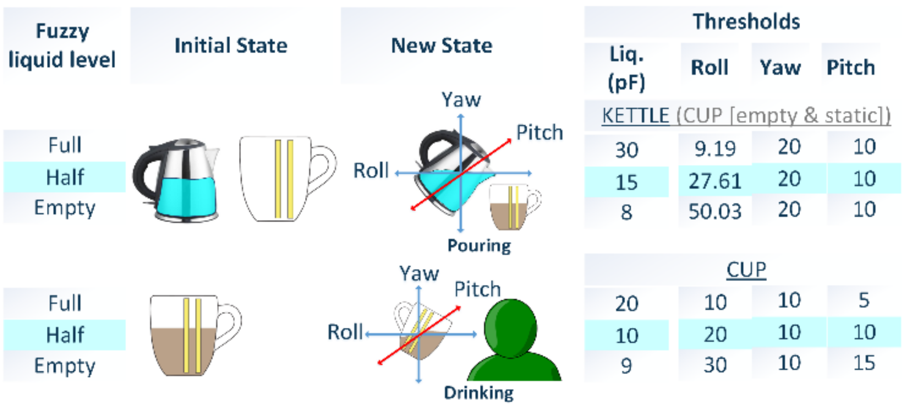


Figure 6.6. Detecting incomplete actions by combining position and water level sensor to detect "*pouring*" actions

Table 6.2. Example of detecting incomplete actions using fuzzy reasoning for pouring and drinking actions.

S	Conditions
1	[kettle=on→kettle=off, kettle_liquid_pf=!empty, cup_liquid_pf !=full, kettle_obj_temp>=hot **OR** very hot]
2	Kettle pouring: - **IF** liquid_pf >= full (30), **THEN** at least roll, yaw, pitch till half threshold amount (9.19, 20, 10). - **IF** liquid_pf (15) <= half (15), **THEN** at least roll, yaw, pitch till half threshold amount (27.61, 20, 10). - **IF** liquid_pf <= empty (8), **THEN** at least roll, yaw, pitch till half threshold amount (50.03, 20, 10).
	Cup filling: - **IF** liquid_pf >= full (20), **THEN** at least roll, yaw, pitch till half threshold amount (20, 10, 5). - **IF** liquid_pf(10) <= half (10), **THEN** at least roll, yaw, pitch till half threshold amount (20. 10. 10). - **IF** liquid_pf <= empty (9), **THEN** at least roll, yaw, pitch till half threshold amount (30, 10, 15).
3	[kettle=off, liquid_pf<=half, cup_liquid_pf!=empty, cup_obj_temp **AND** kettle_obj_temp >= hot **OR** very hot]

Note: S: state, (1): prior, (2) present and (3) post

6.3.2.3. Uncertainty Reasoning at Activity and Action Level

The activity reasoning results from task one and two are used as evidence for probabilistic reasoning at the activity and action level. Therefore, SSBN is created based on the MTheory and MFrags defined in the PR-OWL model and evidences are mapped to relevant probabilistic distribution table which contains possible states and probabilistic values of occurring. These states include Boolean values in the PR-OWL and assumes fuzzy states of the sensor/object is reasoned in task two of the reasoning process. For instance, battery level of a given sensor is measured as 10%, the result from fuzzy reasoning in task two would be "*low*" and this low value will be used as a piece of evidence for a given *po2:DeclarativeProbability* table, i.e., change probability value for the state of "*low*" to 100% and others possible values in the table to 0%. The effected probability tables which depend on the sensor to function correctly in the SSBN will be propagated. Likewise, the collection of pieces of evidence is applied to the SSBN and the uncertainties defined at action/activity levels are calculated.

6.3.3. Algorithm for Single-user HAR with Fuzzy and Uncertainty Knowledge

Table 6.3 presents a pseudocode algorithm for performing activity recognition at multi-granularity action level using fuzzy and uncertainty knowledge model. More specifically, detecting missing sensors using Allen temporal rules, incomplete actions using fuzzy knowledge for a given ADL and handling uncertainties by propagating the pieces of evidence in SSBN.

The algorithm takes in segmented sensors (*segmentedSensors*) and candidate ADL class (*candidateADLClass*) based on T-Box reasoning as input and iterates over each sensor to analyse the data. The algorithm assumes that the sensors data are segmented, filtered to overcome any noise and drifting in data over time and preliminary result of candidate ADL class is performed by Pellet reasoner using T-box and A-box reasoning techniques. The algorithm has three main components.

Table 6.3. Pseudocode for single-user AR which detects missing and incomplete actions in a given ADL with their uncertainties.

```
Input: segmentedSensors, candidateADLClass
Output: arResult
1  List<> depsSeqDone = new list<>();
2  List<> completedFineActs = new list<>();
3  for(Sensor s :  segmentedSensors)
4     List<> allMActsSensors = getAllMActs(candidateADLClass);
5     List<> allMDeptsSeq = getAllMDepsSeq(candidateADLClass);
6     //1) FINDING MISSING SENSORS (Course-grained AR)
7     if(allMActsSensors.contains(s))
8        Boolean r = AllenRulesCheckerUtils.run(allMDeptsSeq, depsSeqDone, s);
9        PROWLResUtils.updateSSBN(candidateADLClass, s, r); endif
10    //2) FINDING INCOMPLETE ACTIONS (Fine-grained AR)
11    if(!completedFineActs.contains(s))
12       Map tv = getFuzzySensorThresholds(candidateADLClass, s));
13       FuzzyDLResult fr = FuzzyDLUtils.run(tv, s.getObservedValues(3));
14       if(fr!=null)
15          completedFineActs.add(s);
16          PROWLResUtils.updateSSBN(candidateADLClass, s, fr);
17          arResult.fuzzyResult(fr); endif
18    //3) PROPAGATING overall SSBN
19    arResult.prowlResult(PROWLResUtils.propagation(candidateADLClass));
20 endfor
```

In the first component, lines 1-9, list and Boolean variables are defined to detect missing actions for a given ADL class. The *depsSeqDone* and *completedFineActs* list variables on lines 1-2 are used to record the sequence of mandatory and fine-grained actions for a given activity are complete. On line 3, the for loop iterates over each of the sensors in a *segmentedSensors* for a given activity to detect missing, incomplete actions and perform uncertainty reasoning. The for-loop initially retrieves all mandatory actions using *getAllMActs* function for a *candidateADLClass* and store it in *allMActsSensors* list on line 4. Similarly, on line 5, all the dependency sequences of actions required for a given activity are retrieved by *getAllMDepsSeq* function based on the *candidateADLClass* and stored in *allMDeptsSeq* list. These *getAllMActs* and *allMDeptsSeq* list are used to first check if the given sensor (*s*) in *segmentedSensors* is part of *getAllMActs* by using if statement on line 7. Next, line 8, Allen rules are used to identify missing action in the *allMDeptsSeq* using *AllenRulesCheckerUtils.run()* function and store them in the *depsSeqDone* list. The *AllenRulesCheckerUtils.run()* function returns a Boolean flag stored as *r* variable to indicate if the *s* is part of the missing mandatory actions. This *r* is then passed to probabilistic reasoner on line 9 to create/update the SSBN for a given *candidateADLClass* along with the metadata about the sensor to be used for uncertainty reasoning, i.e., to determine total active sensors on the network, object it is attached to and its functional property.

The second component lines 10-20, incomplete fine-grained actions for a *candidateADLClass* using *s* is detected. The if condition on line 11, checks if fine-grained action related to the sensor is not already detected. For illustrative purpose, it is assumed one sensor is only used to detect one action. If the fine-grained action for related to *s* has not been conducted previously, fuzzy reasoning is performed using *FuzzyDLUtils.run()* function by initially getting action's fuzzy rules and thresholds Map values using getFuzzySensorThresholds function on lines 12-13. The result from the fuzzy reasoning is stored in the *FuzzyDLResult* class with the variable *fr*. If the *fr* is not null, line 14, it means the fine-grained action with *s* is successfully completed and stored in *completedFineActs* list on line 15. SSBN is updated to indicate completion of the action in the uncertainty model using *PROWLResUtils.updateSSBN()* function on line 16. In addition, *fr* is updated into the *arResult* output variable on line 17.

Finally, the third component, lines 18-20, perform uncertainty reasoning using *PROWLResUtils.propagation* function based on pieces of evidence provided regarding the *s*, detected actions and object's functional properties. The SSBN propagation result containing uncertainty estimation of completion of a *candidateADLClass* is stored *arResult* and output to the user.

6.4. Evaluation and Discussions

To illustrate the applicability and accuracy of the semantical-enabled HAR framework, a case study to recognition *MakeTea* ADL and hot water pouring action from the kettle to the cup with relevant uncertainty factors is presented in Figure 6.7. As per the framework, basic context/entities around the environment is initially created. In this case, the crisp OWL contains *MakeTea* class, a subset of the ADL class, requires *Kettle* as one of the utensils (*hasUtensils*) located in the *Kitchen* to complete the activity. To monitor the user interaction with an instance of the kettle (*blueKettle*), two sensing platforms are described: *ESP8266_12E* (which is a miniature WIFI-enabled microcontroller) and *BLE_TI_SensorTag* (TI sensor tag with BLE communication protocol). These platforms are described using SSN/SOSA vocabulary's class and properties. Both of the sensing platforms *sosa:host* one or more sensors with different types of sensing attributes such as *temperature*, *accelerometer*, *gyroscope* and *liquid level*. The instances of two microcontrollers, MC_1 and ST_1 and instances of sensors types are created and interlinked with *sosa:host* relationship. In order to store the four *sosa:Observations* of these sensors at t_1-t_2, instances, lq_t_1, t_t_2, a_t_2, and g_t_2 are created with their respective data using *sosa:hasSimpleResult* data property.

These sensor observation values stored in *lq_t1*, *t_t2*, *a_t2* and *g_t2* are assumed to contain *hasFuzzyValue* data property defined as the observation property. Hence, in the fuzzy owl model, fuzzy states (*KettleLiquidLevelLow, KettleLiquidLevelMinimum, …,* *KettleLiquidLevelHigh*) are defined as classes for the sensor type (*LiquidLevel*) attached to an object (*KettleLiquidLevel*). The respective membership functions for each fuzzy state is defined as datatypes using fuzzyDL plugin in Protégé. Next, these fuzzy datatypes from different sensor attributes are used to create fuzzy rules (*KettleHotWaterPour*, *KettleHotWaterPourToCup*) to create a condition under which kettle is pouring hot water in the cup. This enables fuzzyDL reasoner to determine if the sensor inputs belong to a given rule and if the action was conducted successfully.

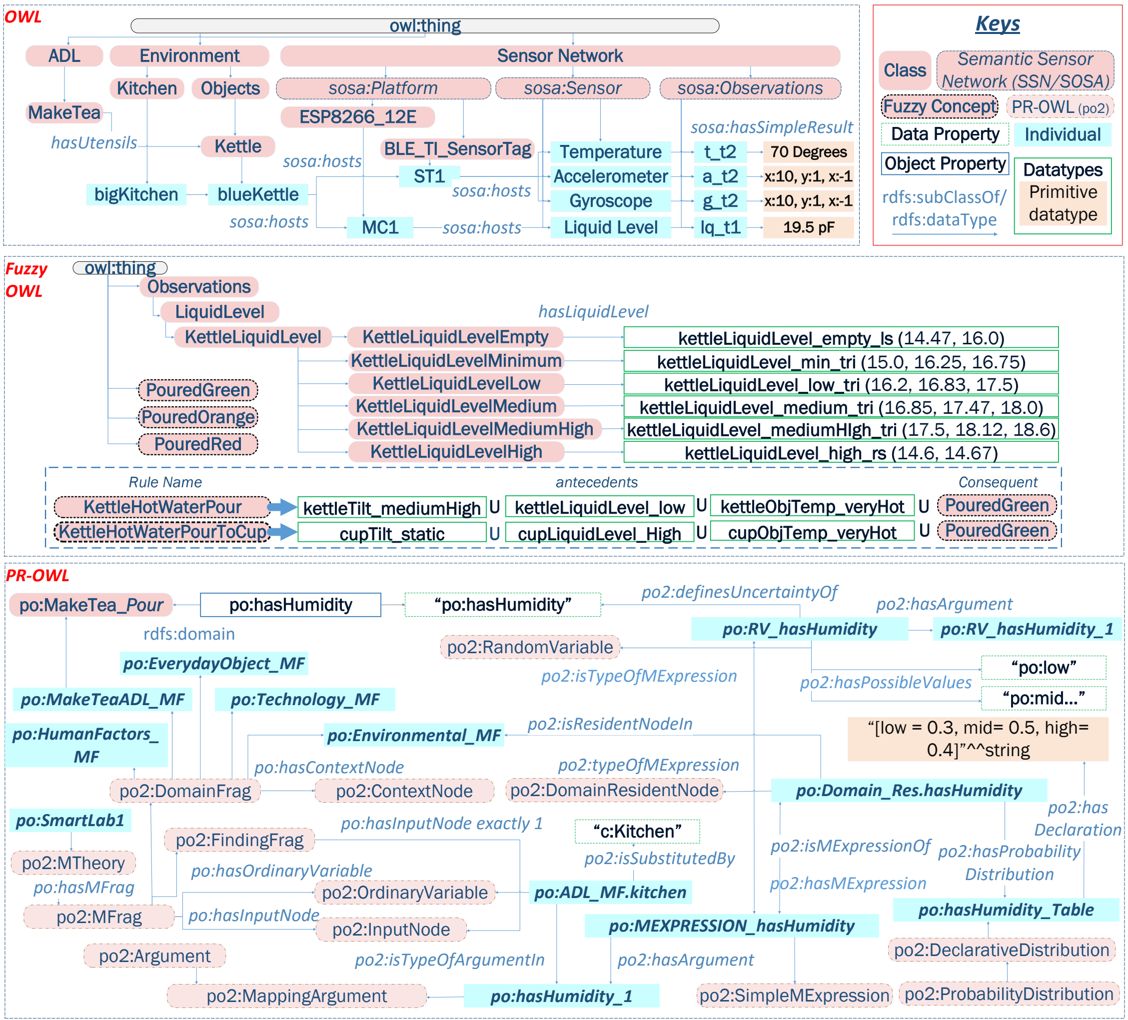


Figure 6.7. Case study: *MakeTea* ADL recognition with fine-grained pouring action and four uncertainty factors using semantical ontological modelling framework.

The PR-OWL model shows how humidity (*po:hasHumidity*) in the kitchen (*po:ADL_MF.kitchen*) environment creates uncertainty in the activity (*po:MakeTeaADL_MF*) reasoning result and also individual actions such as pouring(*po:MakeTea_Pour*). The *po:RV_hasHumidity* random variable defines possible states of humidity level, i.e., low, medium-high. The *po:hasHumidity_Table* is a *po:DeclarativeDistribution* table which defines the probabilities of humidity state occurring. Upon the evidence collected from humidity sensor and reasoned with fuzzyDL reasoner, the value for the humidity state will be changed to 100% in the SSBN and affected MFrags will be propagated.

The main advantage of this ontological modelling and reasoning framework is that crisp, imprecise concepts and uncertainties of events is supported with improvement in human-machine readability of knowledge. Due to the separation of these three types of domain knowledge, this framework enables the model to be decoupled, lightweight and reusable. Hence, making the model easier to maintain and track the changes over-time. On the contrary, misusing or incorrectly using the classes described at different level abstraction can create undesired duplications and create fragmented knowledge which can be challenging to comprehend. Thus, leading to creating inefficiency in AR modelling and reasoning process. Hence, the domain experts are required to carefully map the right entity or concept with their appropriate characteristics that are conceptualised at the appropriate abstraction level of interest.

The critical limiting factor of this framework is the lack of compatible tools to model crisp, fuzzy and imprecise knowledge on a single platform. Hence, influencing AR tasks to introduce additional mapping mechanisms and create complexity in eliciting, conceptualising and reasoning with the knowledge. Another factor inherent to ontological based approach is the requirements of high computation power to analyse the intricate and multi-layered knowledge model on a cloud computing platform. Appropriate parallel computing processes are required with dedicated tasks between the slave computers and their results are synchronised with the master application-level computing processes. Moreover, the opportunity to reduce and offload tasks to edge devices (i.e., the sensing device or device physically close to sensors) and shared device/computers on the same local area network (LAN) require further research to optimise and allow ontological based approaches to test their boundaries.

6.5. Summary and Future work

This chapter investigated in developing a semantical framework that supports factual, imprecise and uncertain knowledge of the real-world when performing single-user AR. The finding from the literature review suggested that studies in the past investigated these three types of knowledge in isolation. However, all three types of knowledge are required in order to handle subjective nature of imprecise sensor measurement to determine the state of an object and probabilistic values for predicting the uncertainties that may occur in the future. In addition, conceptualising and reasoning with the complex relationship between environment, entities, ADLs and sensor network at multiple levels of abstractions and AR at coarse/fine-grained action granularity levels using in a single model proved to be challenging. Therefore, this chapter developed a framework that created a multi-layered knowledge modelling and reasoning processes that bring together crisp, fuzzy OWL and PR-OWL ontologies. In addition, the framework integrates Semantic Sensor Network (SSN) vocabulary in the modelling process to comprehensively describe the heterogeneous sensing platforms with their observation properties, sampling procedure, data storage procedure, system deployment conditions, etc. However, a subset vocabulary of SSN, SOSA (Sensor, Observation, Sample, and Actuator) is available for a lightweight and rapid domain-specific knowledge development.

Overall, the proposed framework is organised using the microservice system architecture (MSA) to perform AR tasks. One of the main benefits of this ontological-based framework is flexible for the model to not only evolve over a period of time using DD approaches but also handle imprecise sensor data and uncertainties of events. Additionally, this framework is flexible and capable of being applied to other domains such as sleep monitoring, healthy eating, intrusion detection and safety risks. However, the main drawback of this framework is that it requires a mapping mechanism between these three models due to incompatibilities of the tools available. This mapping mechanism relies on ontology engineers/developers to use explicit URI definition for the classes/properties with *owl:equivalenceClass* property and carefully selecting the duplicate class in each ontology as the type of descriptions modelled varies when the developing AR algorithm.

Future work will involve developing the framework and comparing the single-user AR accuracy and performance using a multimodal dataset. Although the ontological-based based solutions for HAR tasks demands high computation resources, opportunities created by computing paradigms such as edge and fog computing will be investigated. The main goal of both of these paradigms is to not only delegate tasks and utilise the processing capabilities of the devices closer to the sensing devices but also reduce delays, network traffic, loss of data and overall real-time application of the system.

CHAPTER 7. MULTI-USER ACTIVITY RECOGNITION IN SHARED SMART ENVIRONMENT

A single-user and coarse-grained action level AR have been extensively studied in a smart lab environment, but in the real-world scenario, multiple users share the same space to conduct activities of daily living (ADL). This chapter presents a semantic-enabled approach for multi-user AR and estimating AR confidence level ($\mathcal{ARCL}$) based on pieces of evidence collected and analysed for each ADL at the coarse and fine-grained action level. Firstly, the single-user AR framework is leveraged to encode belief-based importance values for estimating the likelihood for completing the activities at twofold: coarse-grained confidence level ($\mathcal{CCL}$) and fine-grained confidence level ($\mathcal{FCL}$). $\mathcal{ARCL}$ takes the segmented sensor observations, candidate activity classes, importance values specified for critical actions and contextual attributes for a given ADL in the model to calculate $\mathcal{CCL}$ and $\mathcal{FCL}$. $\mathcal{CCL}$ algorithm extracts and takes location, key objects and time interval attributes into consideration whereas, $\mathcal{FCL}$ inspect user's interactions between objects to detect fine-grained actions using predefined thresholds. Secondly, multi-user AR ($\mathcal{MAR}$) approach is proposed to detect, identify and associate user's actions with time-series analysis/location information and discriminative sensor data. More specifically, $\mathcal{MAR}$ binary sensor observations enable detections of multi-user actions using timestamp and location information for time-series analysis. Moreover, fingerprint and RFID tag-based received signal strength indicator (RSSI) sensor information is fused to accurately identify users and associate their actions with the proximity of the object user is interacting. A fusion of ambient sensors and embedded sensors for a non-invasive and non-obstructive data collection approach is proposed and applied to a kitchen and living room application scenario to illustrate its use.

7.1. Introduction

Although extensive work has been carried out to recognise single (predominantly) and multi-user activity, several challenges and restrictions remain unresolved from technical, social and privacy perspectives. CHAPTER 6 developed a single-user AR framework to recognise mixed activities. However, it is assumed that a single-user will use the system in real-world. Consequently, this chapter builds on the single-user AR framework developed in CHAPTER 6 and focuses on highlighting key challenges and developing an approach to estimate multiple users actions at multi granularity level in a shared smart home (SH) environment.

In a shared environment, a single user can perform ADLs by themselves (initially discussed in section 2.1.1), collaboratively with other users or in independently in the same shared environment (i.e., parallel)[28]. The key challenges being focused in this chapter is to estimating AR confidence level ($\mathcal{ARCL}$) for a single and multi-user actions in a shared environment with appropriate sensing parameters. Within a multi-user collaborative AR context, detecting, identifying users and the associating to individual user actions in a given activity is one of the key challenges faced in shared users space [29]. The application for multi-user AR system is to provide personalised assistance to the users and monitoring the overcrowded area in a dwelling and notifying care assistant if the number exceeds a given threshold.

In past studies, diverse sensor-based methods have been used for collecting data from the SH environment based on application requirements. Figure 7.1 depicts what types of data can be retrieved to detect single/multiple users, their purpose and the appropriate ambient, embedded and wearable sensors methods. The information gathered from the smart environment is used to achieve both coarse and fine-grained AR. The coarse-grained AR involves understanding the user(s) generic context such as where the activity is being performed, what objects they are interacting with and what ADLs this action is related to. The ambient and dense sensing methods can provide such information. For instance, a user enters the kitchen opens the cupboard to take out a cup, sugar, tea jar and use the kettle to heat the water. There can be a door or passive infrared sensor (PIR) sensors to detect user location and embedded sensors such as capacitive touch sensor on the door handle. A given AR system can infer these actions to be related to making tea ADL. However, this information is limited and assume fine-grained tasks such as "*pouring*" hot water from the kettle to cup. In addition, the user could be "*tidying up*" the kitchen by putting the objects into their respective places or interleave with another activity by using hot water from the kettle to make "*pasta*" or "*rice*".

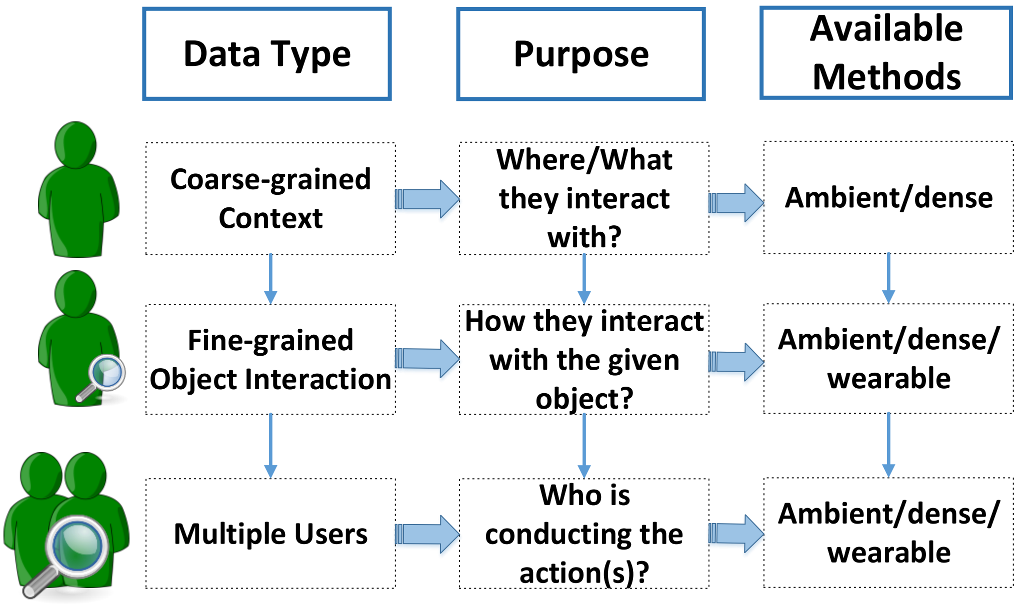


Figure 7.1. Fine-grained multi-user activity recognition approach based on data collection methods

To achieve fine-grained AR, how and when a user interacts with an individual object needs to be analysed with multi-modal data [167]. Previous studies have used inertial measure unit (IMU) position sensors (i.e., accelerometer and gyroscope sensor) either by positioning on the everyday object or using wearable devices with IMU sensors to analyse the position of the everyday objects to infer fine-grained actions such as "*pouring*", "*cleaning*", and "*washing-up*". Once the fine-grained AR is achieved from a set of sensor observations, the next challenge is distinguishing which user and how many people collaborated to conduct activities in the shared space.

The remainder of the chapter is arranged in as follows. Section 7.2 covers related work in detecting single user action at coarse-/fine-grained and multiple users' activities. Section 7.3 proposes a novel approach to estimate and associate multi-user AR challenges with the algorithm details in section 7.4. The approach is applied to a case study with multiple users performing mixed kitchen-based activities in Section 7.5. A discussion is provided in section 7.5.1 and the conclusion with future work in Section 7.6.

7.2. Related Work

To achieve single-user AR at the fine-grained action level, work in [177] joint acceleration, acoustic and multi-sensor classifiers and evaluated it using popular machine learning algorithms; J48 decision tree, random forest, a Bayesian network, and support vector machine. A single off-the-shelf smartwatch was used to sense and reason with the data. The evaluation result indicates that the combined approach achieved greater accuracy (91.5%) in contrast to individual classifiers in recognising five ADLs; eating, vacuuming, sleeping, showering and watching TV. The limiting factor of this approach is that training data is required for individual users and cannot be easily reused. Additionally, low-energy capacity smartwatch must always be worn, which creates practical challenges such as regularly recharging the smartwatch, hindering natural body movements and adoption of smartwatch amongst reluctant users i.e., elderly population. Consequently, the wearable sensors are now being integrated into our clothes and accessories to monitor attributes such as physical movement and posture-based by placing sensors on a different part of the body unobtrusively.

Another study in [170] explores multimodal and multi-positional sensing approach to detect fine-grained actions. Multiple wearable sensors were positioned on a different part of the body and Bluetooth-based beacons to perform AR using the conditional random field (CRF) and decision fusion classifiers. Likewise, work in [178] used inertial ring and bracelet to achieve fine-grained level AR based on the wrist and index finger gestures of eating, drinking and brushing actions with different types of objects. The main limitations for both wearable sensor-based fine-grained AR studies are the lack of semantic reasoning, adaptability, scalability, practical usability, and power consumption challenges. In contrast, work in [215], presents a hybrid method where ontology and Markov Logic Network (MLN) approaches are adopted to permit semantic and probabilistic reasoning amongst activities, context data and sensing devices. The proposed unsupervised approach outperformed the standard supervised method using CASAS and SmartFaber datasets. However, this hybrid approach assumes that action observed by a sensor (mostly binary) has been completed successfully.

In the context of multi-user AR, work in [226] adapted coupled hidden Markov models (CHMMs) by adding vertices to model single and multi-user collaborative activities. The CASAS multi-user dataset was used from Washington State University (WSU) with non-obtrusive sensors consisting of 15 ADLs conducted by two users. However, the approach assumes that the two users in the same region are always performing the collaborative activity and it falls short in distinguishing which user is performing what actions. Another work, [227], predicts next activity in a multi-user smart space using natural language processing (NLP), long short-term memory (LSTM) network and k-means clustering to find a semantical relationship between multiple vectors. The study achieved 85% success rate in recognising activities in a smart meeting room using ambient sensors and actuators. The limitation of this approach is that it cannot detect a total number of users, fine-grained action level activities and when applying the approach in other silent/noisy shared space. In addition, higher window size and predicted activity candidates are required in order to achieve greater accuracy.

Alternative work in [228] presented a method of identifying collaborative and group-based activities using a decentralised approach where wearable sensors and mobile phone were used to perform classification. The information passed from each user's mobile phone are exchanged and analysed for detecting collaborative and independent multi-user activity. The approach further assesses the energy consumption and recognition accuracy using the decentralised method. The single-user activity classification results from a smartphone were shared with other users in the environment in order to detect any collaborative/parallel activities. Similarly, work in [229] tackled challenges of recognising fine-grained and collaborative activities performed by surgeons and support staff in a medical operating theatre setting. The approach leveraged using conditional random fields (CRF) classification method and simulation data from wearable and dense sensors.

In general, recent studies have relied on wearable devices in the context of single-user AR and $\mathcal{MAR}$. However, they have recognised the need to use nonintrusive sensors to monitor users' behaviour and develop real-world applications [29]. For instance, work in [230] identifies a user by using a biometric signature from skull bone conduction using eyewear like google glass. Likewise, work in [6] leveraged wall-mounted radio frequency (RF) transceivers and IR sensors to fingerprint individual users and Gaussian Mixture Model (GMM) for classification. Three test subjects were used, 2 male and 1 female to collect over 2300 labelled samples per subject over 5 days and achieved 83% and 98% accuracy, respectively. However, this approach was tested on a single user at a time and struggled to classify two people with a similar build in stature. In other domains, less intrusive sensors such as fingerprint sensors and voice recognition are commonly being used to identify and authenticate individuals. For instance, smartphone-based attendance and payroll-based systems for employees working remotely [231] and fingerprint sensor based door access control in [232]. However, little has been in explored using these sensors for association sensor observations to a given user for the goal of AR and service provisioning. Nevertheless, fingerprint sensors are subjected to an identity thief, the appropriate positioning of the sensor and may also not be economically feasible to be disturbed in the SH environment, especially, perishable goods. Consequently, work in [233] proposed RFID reader gloves and RFID tags to be placed on objects of interest to detect object interactions. This work has shown the promising results in associating user interaction with objects in a shared environment. Yet, there is a constraint of wearing RFID reader glove all the time to sense tags which is potentially the least desirable solution for an elderly or disabled people in terms of practical use[130].

Alternatively, fixed location ultra-high frequency (UHF) RFID tag reader and passive RFID-tag are used to detect and track multiple users using tag ID and RSSI values within the smart environment. For instance, work in [234] presented SmartWall with passive tags attached to the wall and a single UHF RFID tag reader to recognise mixed activities conducted in the shared environment based on object occlusion method. This approach yielded in high accuracy of recognising 12 activities such as sitting, standing and falling within a 5-meter radius from the SmartWall in comparison to the random forest, logistic regression and SVM classifiers. However, this approach has only been tested with a single user and may require further investigation to support multi-user AR. A similar study [179] developed a passive RFID based Moo Tag with onboard 3-axes accelerator sensor to improve the accuracy of action detection. Nevertheless, these studies show limited support to associate user’s interactions of objects in a shared environment. Furthermore, it still requires high installation cost for with RFID tags across all the walls/everyday objects (non-/perishable) and constant energy requirement for UHF RFID readers with a long-range antenna.

In this chapter, the use of KD modelling and classification techniques for ADLs is explored using sensor-based data collection methods to achieve multi-user AR at the coarse-/fine-grained action level. In addition, for practical and real-world applications, non-wearable sensing approach is presented using general ambient and object embedded (dense) sensing method for this goal.

7.3. Multi-user AR within Shared SH Environments

A multi-user AR is developed to analyse a set of sensors data semantically segmented relevant to ongoing activities with discriminative sensor-based approaches to associate user actions estimating overall $\mathcal{ARCL}$. $\mathcal{ARCL}$ is calculated based on importance weighted values embedded into the ontological model and discriminative sensor data to help identify the users in a given environment.

An ontological based activity modelling and reasoning approach are utilised to conceptually describe concepts, relationships and instances to formally define ADLs with environmental objects that have sensors attached to them. Thus, incoming sensor events are initially semantically evaluated and separated in a set of sensors based on the relationships with the ongoing ADL(s). The segmented sets of sensors for a given ADL are then evaluated at two granularity levels: coarse- and fine-grained. At the coarse level, three key context satisfactory criteria are evaluated from the sensor's relationship with a given ADL: location (L_r), key objects (KO_s) and time interval (TI_t). Whereas, at fine-grained granularity level, detection of key fine-grained actions (FA_i) with the sensor data is performed with a specific object and matched against thresholds.

The ontological based AR approach by itself cannot distinguish who is performing the actions and how many people are collaborative or independently performing activities in the shared space. Therefore, discriminative sensing approach and pattern detection techniques are required that can identify collaborative activity occurring, individuals with their unique signatures and track their activities at the action level. Details of proposed multi-user AR is provided in section 7.3.3.

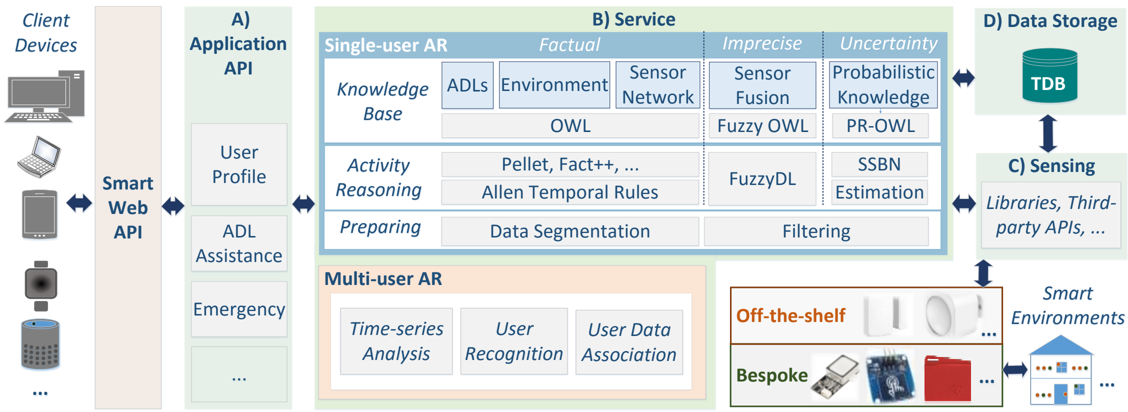


Figure 7.2. Multi-user AR system architecture overview

A multi-layered microservices-based system architecture (MSA) system is proposed and graphically depicted in Figure 7.2 to conduct multi-user AR tasks. This approach enables AR tasks to be delegated between web services, client devices and sensing environment. In the previous multi-layered SOA system and hardware configuration was proposed in [157], [158]. However, this approach required single web service to be deployed on a specialised and powerful machine with multiple core central processing unit (CPU) and several threads to conduct all the AR tasks efficiently. Nevertheless, MSA enables single web service performing all AR tasks to be shared amongst multiple machines. Therefore, the sensing data is centrally collated by the data collection web service. The data collection web service consists of a utility layer with dedicated packages and classes to collect sensor data from multiple sources, store/manipulate data from the database and provide other knowledge reasoning utilities. The data analytics web service put each sensor observations in the queue, semantically segment (more details in section 7.3.1) the queue based on ADL unfolding and interact with data collection web service API. The segmented set of actions based ongoing ADL and their descriptions are then used to perform single-user (coarse- and fine-grained) and multi-user AR. All the AR results and sensor events log are stored in the graph-based database (Apache Jena Fuseki Server). The AR results, sensor events, and other requests made by client devices are made available via *SmartWebAPI* using RESTful communication protocol and provide a response in multiple formats, i.e., JavaScript Object Notation (JSON) and Extensible Markup Language (XML).

7.3.1. Data Segmentation Process

To perform single user $\mathcal{ARCL}$ or $\mathcal{MAR}$, the incoming sensor observations are initially segmented incrementally in twofold. Firstly, the terminology box (T-box) reasoning is performed to check if the given event is part of ongoing ADL description in the ontology, otherwise creates a new activity queue for the first event. These checks consist of performing satisfiability, subsumption, and instance checking using incremental Pellet reasoner. The second step is only executed if there are any conflicts identified by the Pellet reasoner in step one. In the second step, assertion box (A-box) reasoning is performed by querying the triplestore to find relevant ADL preferences specified by the user and check if the sensor observation event is part of any preferences. In the event where both steps find a discrepancy in ADL description and fail to find any association with other ongoing activities, the start of the new activity is assumed. For this purpose, multithreading is used where each thread represents individual ongoing ADL. These ADL threads capture any sensor events based on semantic relevance to the respective activity independently. The comprehensive details on how the generic and preference of users to conduct ADLs are modelled and reasoned for the semantical segmentation can be found in CHAPTER 3.

7.3.2. Single-user Activity Recognition

7.3.2.1. Ontological Knowledge Representation

The ontological activity modelling approach allows the relationship between everyday objects within the living environment and generic ADLs to be logically representation. The crisp knowledge has been conceptualised using formal theories and it allows expressive relationships to be defined between multiple entities. Description logic (DL) is a family of formal knowledge representation languages that are supported by OWL and RDF Schema vocabulary. DL enables the logical representation of conceptual structures and relationships using three main elements: concepts, roles and individuals. The concepts denote to sets of individuals and roles denote to binary relationships between individuals. The individuals are instances of concepts. The vocabulary used for defining concepts and roles of an application domain is referred to as the terminology box or the *TBox* in short. All named individuals are referred to as assertions about a real-world domain or the *ABox*. Hence, DL allows users to build complex descriptions of concepts and roles. Furthermore, DL based reasoners can be used to automatically perform inferencing to derive facts that are not expressed explicitly in the ontological model; this process is known as T-Box reasoning.

The user-specific preferences are also described as instances of a specific ADL class and stored in the graph-based database (triplestore). This process is known as A-box reasoning. To avoid conflicts between instance checking and ADL class satisfiability for user's preferences, generic object relationship is used. SPARQL Protocol and RDF query language (SPARQL) is used to retrieve relevant user's preferences. Both generic and user-specific preferences knowledge are utilised to segment each sensor observation into a relevant set of activity queues and then perform further activity classification.

7.3.2.2. Multi-granularity ADL Description

The environmental objects, ADLs, sensing network and their relationships are modelled using ontology editor (i.e., Protégé). Each ADL is further described with three coarse-grained parameters (L_r, KO_s and TI_t) and key fine-grain actions (FA_i) performed with a specific object. The coarse-grained parameters are selected to check if the key actions for ADL are performed at most, during the appropriate time of the day and place.

The multi-granularly descriptions of ADLs are given importance values defined by domain expert's knowledge. Therefore, as the activity unfolds for a given ADL, their importance values are accumulated and averaged out with the number of parameters to calculate coarse-grained confidence level ($\mathcal{CCL}$) and fine-grained confidence level ($\mathcal{FCL}$). $\mathcal{CCL}$ takes L_r, KO_s and TI_t attributes from the sensor data and ADL description into consideration. Each parameter is given importance values (total of 100%) that are defined by a domain expert in the ADL knowledge base for individual ADL. Hence, the sum of the weighted values of three coarse-grain parameters are calculated and then averaged out to calculate $\mathcal{CCL}$; see equation 7-1. Similarly, $\mathcal{FCL}$ analyses sensor data of the individual object to detect key FA_i and add all respective importance values (total of 100%); see equation 7-2. The values are of $\mathcal{CCL}$ and $\mathcal{FCL}$ are combined and then averaged out to get an overall activity recognition confidence level ($\mathcal{ARCL}$) for a given ongoing activity. In addition, the assurance of detecting fine-grained action rather than assuming that a given action has taken place, the $\mathcal{FCL}$ value is given three times the importance than $\mathcal{CCL}$ value as shown in the equation 7-3.

$$\mathcal{CCL} = \left(\frac{\sum_{r=1}^{x} L_r + \sum_{s=1}^{y} KO_s + \sum_{t=1}^{z} TI_t}{3}\right) \qquad \textbf{7-1}$$

$$\mathcal{FCL} = \sum_{i=1}^{n} FA_i \qquad \textbf{7-2}$$

$$\boldsymbol{\mathcal{ARCL}} = \left(\frac{\mathcal{FCL} * 3 + \mathcal{CCL}}{4}\right) \qquad \textbf{7-3}$$

Table 7.1. Spatio-Temporal ADL description for multi-granularity AR

Activity	**Coarse-grained parameters**			**Fine-grained Actions (FA) ***
	*Location (L) **	*Key objects (KO) **	*Time Int. (TI) **	
MakeTea	Kitchen	TeaBag(50), Cup(10), Kettle(30), WaterTap(10)	6.30-11.30am, 3-6.30pm	Pouring (50), Drinking (20), Filling(30), WashingUp(10)
MakeBaked Beans	Kitchen	BakedBeansCan(50), MicrowaveBowl(25), Microwave (25)	6.30am-2.30pm, 6-8.30pm	ToasterOn(70), MargarineSpread(30)
MakeToast	Kitchen	Toaster (50), BreadSlice(30), Margarine(10), EatingKnife(10)	6.30-11.30am, 3-6.30pm	CanOpening(60), CanPouring(20), TransferringFood (20)
Take Medicine Dose	Kitchen (50), Living room (50)	MedicineBox(80), WaterTap(10), Glass(10)	8-10am,1-2pm, 5-7pm, 10-11pm	Eating/ Drinking Medicine (70), DrinkingWater(30)
Tidying	Any room/ Unspecified	Bin (25), Sink (25), Furniture (50)	Unspecified	MovingObject(40), PutIntoSink(20), CloseKitchen Furniture (20), PutInBin(20)
Washing Up	Kitchen	WashingSoap(30), HandGloves(5), EatingCutlery(20), CookingCutlery(20), WaterTap(25)	Unspecified	Wipe (35), CircularMotion(35), WashingLiquid(30)

Note: * total importance weighting of 100% per activity unless stated.

For example, to *Make Tea*, users can normally perform this task during the morning or afternoon by going in the kitchen and must interact with KO such as a *cup*, *kettle*, *water tap* and *tea bag/jar* to complete the activity. These KO are given importance value to determine the level of completion of the activity. The importance values are derived if the actions between KO are shared and how significant it is for the action to be completed a given activity. For instance, the interaction between *TeaBag* and *Kettle* can be more important than *Cup* and *WaterTap* in order to determine the action of M*ake Tea* activity. Similarly, other activities can be described as illustrated in Table 7.1. In the case where L and TI are shared for a given activity, the total importance values available (100%) are distributed as illustrated for *TakeMedicineDose* activity.

7.3.2.3. Sensing Attributes

The generic context can be obtained when a user opens the *kitchen door*, interacts with s*ugar jar, tea bag, cup, water tap* and *kettle*. Although these actions belong to "*Make Tea*" ADL, it does not necessarily mean the user has complicated the action or they could be performing more generic "*tidying*" activity. Therefore, to achieve fine-grain AR, each object interactions and usage must be evaluated to detect key actions such as "*pouring*" hot water from the kettle into the cup. The advancement in sensing technology is becoming cheaper, smaller, wireless and energy-efficient. However, collecting data from pre-installed sensing infrastructure remain costly, difficult to maintain (i.e., battery life), and the position of the sensors can be fixed or portable [8]. Hence, wearable sensors can be more appropriate to monitor vital physiological parts; however, forces one to wear it at all times and create practical challenges [29].

With this in consideration, a fusion of ambient and object embedded (dense) sensors data collection methods is proposed to detect actions at coarse- and fine-grained level. The ambient sensors will provide coarse-grained contextual information about the environment and the objects user is interact using sensors such as motion detector, magnetic door/window and capacitive touch sensors. In contrast, dense sensors such as TI SensorTags for object positioning and liquid level sensing approach are proposed to be attached to relevant everyday objects for fine-grained object usage recognition. For instance, "*pouring*" water from the kettle to a cup can be determined if the correlation between the changing state of the water level and tilting position of the kettle and cup exceeding a given threshold. This threshold can vary depending on the initial quantity of the water, dimensions of the object and the sensor placement on the everyday objects. Likewise, other fine-grained actions such as "*drinking*" from the cup can also be detected with relevant sensors attached to the object. A heartbeat signal and liquid level information of a kettle, cup or other containers can be sent the web service at a regular interval or upon a change in water level detection threshold to reduce the sensor data transmission traffic and energy.

However, not all everyday objects would require water level sensing actions such as "*opening can*" for *MakeBakedBeans* activity and *"transferring food"* to a plate. The fine-grained actions, along with their belief weightings for the other ADLs are listed in Table 7.1. The everyday object and action-specific thresholds can be defined as instances, stored and queried from the triplestore, or logical rules can be specified (i.e., using Semantic Web Rule Language (SWRL) and fuzzy rules [187]). However, with the complex semantical reasoning and computation requirement at sensor segmentation stage, storing and retrieving threshold values based on individual objects in the triplestore would be efficient to reduce computational resources required for runtime rules-based reasoning. Moreover, to detect multi-users' activities, fingerprint sensors are embedded and strategically positioned on to everyday objects. As fingerprint sensors cannot be included to all the everyday objects, smart fabrics with an RFID tag embedded into user's clothes is proposed with UHF RFID reader detecting users in the environment and their location in a shared space. Figure 7.3 depicts the overall sensing parameters and data types required for a single user (coarse-/fine-grained action level) and multi-user AR.

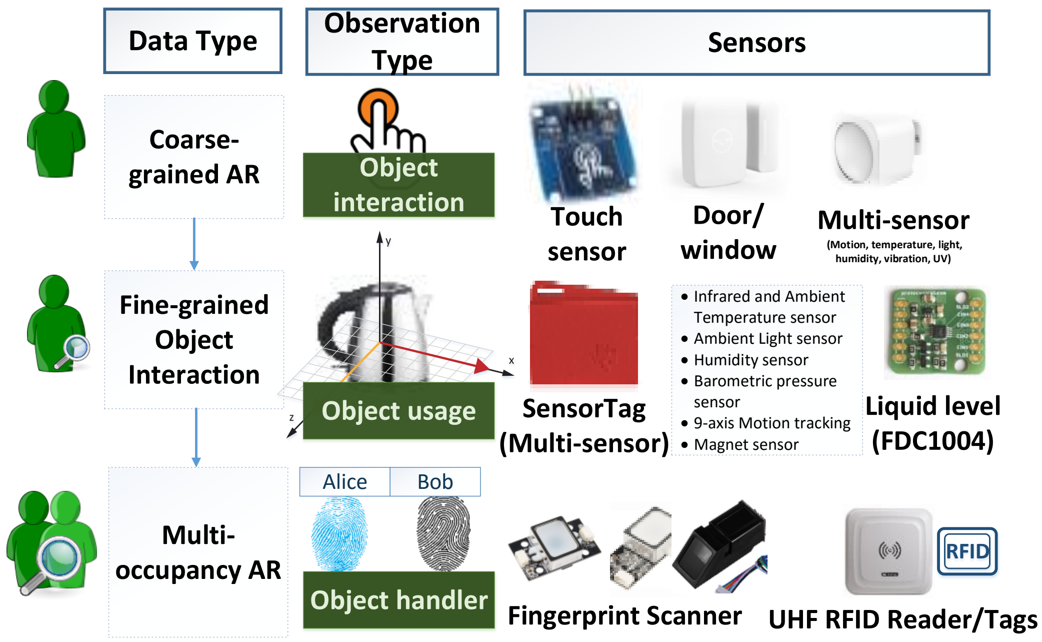


Figure 7.3. Proposed sensing parameters for multimodal single and multi-user AR.

7.3.3. Multi-user Activity Recognition

$\mathcal{MAR}$ analyses segmented set of sensors observations of a given $\mathcal{ADL}_n$ to detect, identify and associate user's (I_j) actions with everyday objects ($KO_s[FA_i]$), contextual and environment sensor data (i.e., L_r, TI_t) as described in equation 7-4.

$$\boldsymbol{\mathcal{MAR}} = \left(\mathcal{ADL}_n\left[I_j[L_r, KO_s[FA_i], TI_t\,]\right]\right) \qquad \textbf{7-4}$$

The detection of multi-user actions is performed by using timestamp information from the number of objects simultaneously interacted in a given time interval and location. A user is assumed to interact or hold no more than two or three objects at the same time interval. For this, fix time windowing analysis is performed to detect potential multiple users-based activities. It is also assumed that a single user cannot be in two locations at the same time. Therefore, in a given fixed time window, if two motion or pressure sensors located in different areas are activated, multiple users are detected. However, both location and time windowing approach falls short in identifying how many users are in a location at the given time interval. Therefore, smart textile with RFID tags and UHF RFID readers deployed in the shared area is proposed. RFID reader range is determined based on the antenna and frequency. The RSSI signal can further help to detect how far a user is from the reader and hence track their approximate location.

Although location/time windowing and RFID reader/tag-based approach can detect multiple users in the same space, it is still unable to distinguish which user is interacting with a given object. Hence, a fingerprint sensor attached to everyday objects is proposed to identify which user in shared space has interacted with the object. The fingerprint sensor can repeatedly scan for a fingerprint and automatically match against the enrolled/stored fingerprints in the sensor's database and provides identify (ID) number. Each fingerprint sensor can internally store fingerprint images (i.e., up to 3000 in GT-521F52) with a unique ID and perform image matching with a low error rate and delays. However, all the users sharing the space are required to initially scan their fingers and thumbs on a fingerprint sensor and synchronise this information with other fingerprint scanners. The unique IDs generated for each finger and the sensor are mapped together and associated user information is stored in the triplestore. Therefore, when the fingerprint sensor is in observation mode, the scanned fingerprint image is matched using inbuilt recognition functionality and the ID matched sent to the central system (i.e., data collection web service).

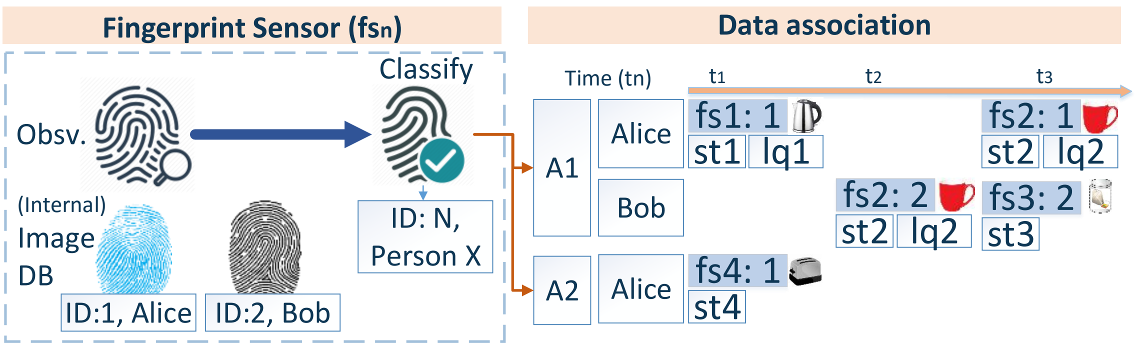


Figure 7.4. Example of identifying multi-users actions and associating their collaborative or parallel actions in a shared space.

Figure 7.4 presents an example of $\mathcal{MAR}$ process during *make tea* (A1) and *make toast* (A2) activities with sensors (fingerprint(fp_n), sensor tag(st_m), and liquid (lq_o)) attached to the *kettle* (fs_1, st_1, lq_1), *cup* (fs_2, st_2, lq_2), *tea jar* (fs_3, st_3), and *toaster* (fs_4, st_4). The detection process initially counts 3 objects interactions for A1 and 1 object for A2 within three seconds(t_n) time window. The count of the objects interacted for A1 exceeds the pre-defined threshold (i.e., < 3) per person within a fixed three-second window. Therefore, the users, Alice and Bob conducting actions for A1 are then identified using fp_n sensors data. Finally, the other sensors attached to 3 objects are associated with users.

7.4. Multi-user AR Confidence Level (ARCL) Algorithm

Table 7.2 presents the algorithm as a pseudo-code divided into four sections to perform $\mathcal{CCL}$, $\mathcal{FCL}$, $\mathcal{ARCL}$ and $\mathcal{MAR}$. The algorithm takes in segmented sensors (*segmentedSensors*) as an input based on inferred ADL and iterates over each sensor to calculate AR confidence level and associated sensor events to the users. Furthermore, relevant parameters and importance values retrieved from the triplestore (TDB) are passed as an input for simplicity when calculating $\mathcal{CCL}$ (*cclKeyObjectsAndWeights*, *cclADLLocationsAndWeights*, *cclADLTIWeights*) and $\mathcal{FCL}$ (*fclSensorsAndWeights*). The algorithm outputs the AR results (*arResult*) containing $\mathcal{ARCL}$ and $\mathcal{MAR}$ containing data association between the user and the everyday objects.

The first section of the algorithm lines 1-9, accumulatively calculates the $\mathcal{CCL}$ value (*cclResult*) by reviewing each sensor observation with the set of key objects, location and time interval passed (lines 5-9) as an input to retrieve relevant importance values for the ADL actions. The average value of *cclResult* value is accumulated by the previous *cclResult* and updated with the new $\mathcal{CCL}$ result (lines 9). Subsequently, the second section calculates $\mathcal{FCL}$ *(fclResult* on lines 10-19) by detecting granular actions using the sensor data, predefined thresholds and the importance values for a given action to calculate $\mathcal{FCL}$. The *detectFineGrainedAction* function takes the thresholds related to a particular action and sensor type in order to compare the observed sensor values and return respective action (lines 14). If the fine-grained action is detected, the action's importance is added to the *fclResult*, timed by three, due to the importance factor, added to the *cclResult* and the average is calculated. The *fclResult* is then accumulated and updated with previous sensor's *fclResult* (lines 15-18). The overall $\mathcal{ARCL}$ value is calculated based on the *cclResult* and *fclResult* as a third step (lines 19-20).

The final part of the algorithm performs $\mathcal{MAR}$ by detecting and associate each sensor events to the relevant user in lines 21-24. The *hasMultiUsers()* function, on line 22, take sensor event and perform time windowing and location-based analysis to detect multiple users in the environment. If the multiple users are detected, the *associateSensorWithUser()* function, on line 22, take sensor event and *arResult* to identify and associate the action to the user. The data association function is depended on the fingerprint sensor attached to a given everyday object and the pre-defined knowledge of other sensors attached to the same objects. In the case where there is no association found between the sensor and user, the sensor is added under a temporary "*unknown*" user for future analysis. The temporary users can be later identified and updated by receiving feedback from known users.

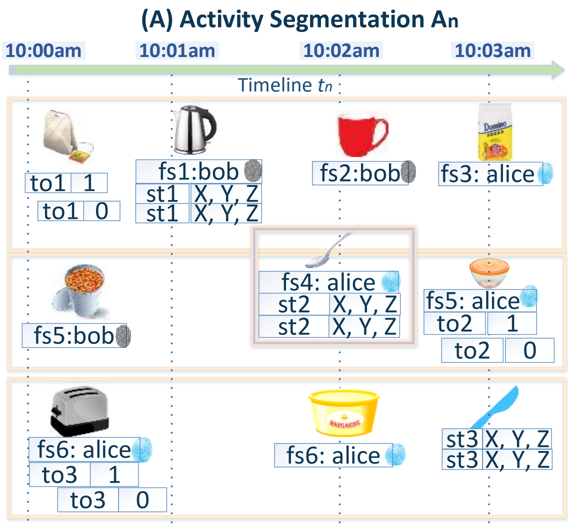
Table 7.2. Pseudocode for calculating $\mathcal{CCL}$, $\mathcal{FCL}$ and $\mathcal{MAR}$ confidence

```
Input: segmentedSensors, cclKeyObjectsAndWeights, cclADLLocationsAndWeights,
cclADLTIWeights, fclSensorsAndWeights
Output: arResult
1    for (Sensor s: segmentedSensors)
2        float cclResult, fclResult, arclResult = 0;
3       //1) course-grained AR and calculating CCL
4       if (cclKeyObjectsAndWeights.contains(s))
5           String location = getSensorLocationFromTDB(s);
6           cclResult += cclADLLocationsAndWeights.getWeight(location);
7           cclResult += cclKeyObjectsAndWeights.getWeight(s);
8           cclResult += cclADLTIWeights.getWeights(s.getTimeStamp());
9           arResult.updateCCLResult(cclResult/3); endif
10      //2) fine-grained AR and calculating FCL
11      if (fclSensorsAndWeights.contains(s))
12          Map w = fclSensorsAndWeights.get(s).getThresholdValues();
13          Map v = s.getDataValue();
14          String a = detectFineGrainedAction(w, v);
15          if (! a.isEmpty())
16              arResult.addFineGrainedActions(a);
17              fclResult += fclSensorsAndWeights.getWeight(a); endif
18          arResult.updateFCLResult(fclResult); endif
19      //3) overall ARCL value
20      arclResult = (fclResult*3) + cclResult/4;
21      //4) multi-user AR (MAR) / data association
22      if (hasMultiUsers(s) & !associateSensorWithUser(s, arResult))
23          arResult.addDataAssociation(new User("unknown"), s); endif
24   endfor
```

7.5. Testing and Evaluation

An ADL scenario is described in Figure 7.5(a) where three mixed activities are carried out in the shared kitchen by two users, Bob and Alice. The actions for three activities, *MakeTea* (A1), *MakePasta* (A2) and *MakeToast* (A3) occurring between 10.00am to 10.03am are illustrated. The sensor observations are collected by the respective event handler classes in *SensingUtils* package of the web service and appended to the observations queue. Each sensor observations occurring at a given time (t_n) are then semantically segmented based on the object's relationship with a set of actions specified in the ADL description and only appended to the activity (A_n) thread if the observed action matches the ADL description.

The $\mathcal{ARCL}$ algorithm is performed in four stages by the individual activity thread and the sample results are depicted in Figure 7.5 (b). The first stage is to perform context analysis of each activity and calculating $\mathcal{CCL}$, i.e., identifying the location, key objects and time interval to calculate the confidence level of the activity occurring. The location information of the everyday object is predefined for fixed objects such as kettle, toaster and microwave. The key objects for each activity, location and time interval are mapped with the importance of a given activity which is stored and queried from the triplestore.



(B) Segmented Observations Analysis

1) CCL	2) FCL	3) ARCL	4) MAR
A1: MakeTea 96.67%	st1=true (pouring) 50%	**61.67%**	alice (x1), bob (x2) Other (x2)
A2: MakeBaked Beans 91.67%	st2=true (transfer) 20%	**37.92%**	alice (x2), bob (x1)
A3: MakeToast 93.33%	st3=false (spreading) 0%	**23.33%**	alice (x2) other (x1)

Figure 7.5. (a) Three ADLs segments processed by ARCL algorithm, (b) four stages of multi-granularly single and multi-user activity detection result.

The second stage is to inspect sensor data to detect if the user has performed fine-grained actions such as "*pouring*" by inspecting accelerometer, gyroscope and liquid level sensor data. The threshold to detect "*pouring*" action vary depending on the dimensions of everyday objects and the quantity of content inside. Therefore, thresholds are predefined for when liquid quantity is low, medium, and high along the degree of rotation/tilt position for each object type. The associated importance values of both stages are used to calculate $\mathcal{CCL}$ and $\mathcal{FCL}$. The final stage is performing $\mathcal{MAR}$ using fingerprint sensors and associating sensor observations to the user identified. In addition, other sensors attached to the same object to the fingerprint sensor is grouped and associated with the user.

7.5.1. Discussion

Despite the scalability and deployment challenges to attach a fingerprint sensor to each everyday object, this approach can identify individuals more discriminatively than passive identification (ID) broadcasting based approaches[235], [236]. For instance, smart clothing with passive RFID tags [237] can be worn by another person or incorrectly assigned and Bluetooth based smart beacon deployed in the environment that are read by the smartphone belonging to another individual. However, RFID tags and beacon are very unobtrusive and passive sensing approach to detect the number of users and triangulated locations [238] in a shared environment and assume the link to a specific user.

One of the limitations of this approach is that each everyday object of interest would require at least one fingerprint sensor in order to associate each sensor observation with a given user. In addition, the traditional capacitive fingerprint sensors can only cover the small area where a user's fingerprint can be scanned; hence, the position of the sensor is important. However, in the recent advancement in ultrasonic fingerprint technology can help overcome these limitations and reduce the cost of the sensors. Ultrasonic fingerprint sensor technology has been under investigation for more than a decade to overcome the poor performance of capacitive fingerprint sensors when fingers are oily, wet and it can easily be spoofed using printed or moulded fingerprint images [239], [240]. Recently, Qualcomm announced advance fingerprint scanning and authentication technology capable of covering a larger area of the display, thick glass and metal surface [241], [242]. In addition, detection of directional gestures, heartbeat and blood flow even when immersed underwater can be used to add layers of authentication and identification of a user. Mobile phone manufacturer such as Vivo has already integrated this technology into their flagship phones and others such as Apple, Samsung, Xiaomi, and OnePlus 5 are expected to follow soon.

Another limitation when adapting a dense sensing approach is that perishable and recyclable items such as soap, plastic bottles and other packaging materials pose scalability, reusability and integration challenges. In addition, the design of the everyday items and size dimensions parameters determine the sensor positions, hence, the varying threshold values.

7.6. Summary and Future work

This chapter developed the course- and fine-grained activity recognition (AR) algorithms and estimates AR confidence level ($\mathcal{ARCL}$). The coarse-grained confidence level ($\mathcal{CCL}$) algorithm extracts location, time and key objects for a given activity along with their respective importance levels from the segmented sensor observations. Each key actions and parameters are given a pre-defined importance value based on the degree of belief for the action required to occur for calculating the confidence level. To recognise granular user actions using a given object, i.e., "*pouring*" water from the kettle to cup, the fine-grained confidence level ($\mathcal{FCL}$) algorithm is introduced which analysis the sensor observation against the threshold values predefined with the importance level information. The sum of all fine-grained action's importance values is considered three times more important than the $\mathcal{CCL}$ value when calculating the overall $\mathcal{ARCL}$.

In addition, Multi-user AR ($\mathcal{MAR}$) algorithm is proposed, which can detect, identify and associate actions of several users performing collaborative or parallel activities. The approach leverages fix time windowing process to detect maximum objects interactions with a pre-defined threshold and multi-location events. Moreover, smart textile with RFID tags and fingerprint sensors attached to everyday objects is used to identify and associate sensor observations to users. However, the key limitation of this approach is the scalability and maintainability challenge to integrate fingerprint sensors in every object wirelessly. The layered microservices-based system architecture (MSA) system and key sensors have been proposed to create a multi-user smart environment. The key sensors include ambient sensors (door/window, PIR, UHF RFID reader) and dense sensors (inertial measurement unit (IMU), fingerprint, RFID tags, and liquid level sensors) for a non-invasive and non-obstructive data collection. The approach is applied to a use case application scenario where mixed kitchen-based activities with multiple users performing collaborative tasks.

The future work will involve implementing and evaluating the performance and accuracy of the proposed $\mathcal{ARCL}$ algorithms with $\mathcal{MAR}$. In addition, optimising AR performance and investigating in activity learning techniques to evolve ADL models.

CHAPTER 8. MICROSERVICES FOR AMBIENT ASSISTIVE LIVING SYSTEM

While most researchers focus on developing accurate AR approaches, this chapter examines some of the system architectural challenges of the Ambient Assisted Living (AAL) systems. A microservices-based system architecture (MSA) is evaluated in the context of AAL to address some of the shortfalls in the predecessor system implementations using off-the-shelf and open-source hardware and software components. MSA brings together system architecture styles and patterns, semantic web technologies, smart home (SH) technologies and artificial intelligence approaches to support real-time context-aware assistance to the users in a shared environment. The system takes some of the key design requirements such as extensibility, reusability, scalability, and maintainability into consideration that can create a foundation to enrich the capability of real-time monitoring, data collecting, processing and accurately recognising mixed activities. In order to validate the proposed architecture, two types of prototypes built using multi-layered service-oriented system architecture (SOA) and MSA are critically evaluated for the applicability of the system in a real-world multi-user shared living environment scenario.

8.1. Introduction

The increasing global ageing population will inevitably create a greater demand on the health care system that is already facing a shortage of resources. AAL system is a technological solution for this modern-day problem. However, many problems related to system architecture, HAR and SH environment need to be solved in order to fully simulate and/or take the role of a care provider or health care professional to a certain degree [3].

CHAPTER 6 and CHAPTER 7 presented novel approaches to recognise single and multi-user activities in a shared living environment. However, this chapter is set within the context of addressing the three levels of system architecture challenges in building an assistive system. These levels are (a) selecting appropriate style and design pattern, (b) considering specific technological and technical requirements for activity recognition, and (c) building and integrating appropriate wireless sensor technologies for providing real-time assistance and monitoring.

The consecutive sections are organised as follow. Section 8.2 discusses related works and existing systems to find their shortcomings. Sections 8.3 and 8.4 present the proposed microservice system architecture and the implementation details of an assistive system, respectively. Section 8.5 evaluate the proposed system and provides some discussions. It must be noted here that the nature of this chapter is not to introduce a new way of modelling or recognising activities, but rather to assess the feasibility of the proposed system architecture. Finally, sections 8.6 summaries the chapter with recommendations for further work, respectively.

8.2. Related Works

8.2.1. System Architecture for AAL Systems

In the past, several assistive systems were implemented. In particular, two prototype assistive systems were implemented to provide AR and assistance features for the elderly or those who have cognitive difficulties in carrying out ADL, namely, the SMART system [35], [243].

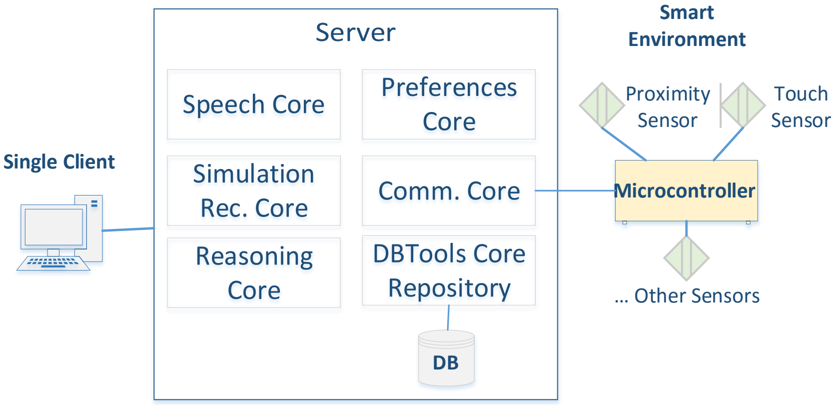


Figure 8.1. System Architecture Overview: Standalone Implementation of the SMART system (2009) [243]

In its initial implementation, the SMART system was built in a standalone environment with a direct interface to the SH environment and featured a rich web-based interface using dotNet programming language. As shown in Figure 8.1, the SMART system consists of six main classes: speech core, reasoning core, preferences core, communication core, simulation recording core, and database tools core. The speech core class is used to output pre-recorded audio messages to the user when the assistance is triggered; personalisation of the pre-recorded message is also supported. The reasoning and user preferences core classes are the core components of this system. The reasoning core class is used to infer the users' activities from their preferences. The user preferences are administered via basic or advance learning methods presented by the system as well as the sensor activation data retrieved from the communication core. The data from the sensor activations (i.e., inferred activities from reasoning) can be recorded using simulation recording core class. Such data can then be exported to the user’s local disk or stored in a repository database as a history log.

In the latter implementation, the SOA approach was introduced (see Figure 8.2) with open-source components. The core system was written in a popular programming language, Java. The main reasons were to move away from a standalone environment as well as to resolve limited community support and proprietary components. This approach allows many users from multiple devices to communicate simultaneously regardless of their operating system. The system further addresses the monolithic code structure of the source code by logically separating it into three web services. The Enterprise Service Bus (ESB) supporting software is used to bind these services together; thus, enabling better maintainability, reuse, and debugging. The system still has a web-based interface that uses JavaScript, Asynchronous JavaScript, and XML (AJAX) features to request and load the data from the ESB. In addition, the Simple Object Access Protocol (SOAP) and Hypertext Transfer Protocol (HTTP) have been used for exchanging data between different devices. Moreover, this service has the potential to be deployed on to the cloud servers that possess the superior computational capacity to perform very complex reasoning within a short amount of time [244]. One of the disadvantages of using this system, however, is that it has multiple web services with an ESB, which requires it to be hosted on the network. This can create unnecessary overhead and delays in the system.

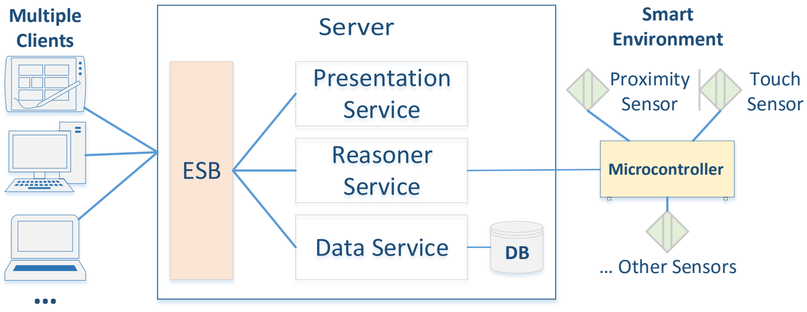


Figure 8.2. System Architecture Overview: Service-oriented implementation of the SMART system (2012) [35]

A previous study [245] presented a location-based context-aware system architecture, in which a range of stakeholders can work collaboratively. The users do not require any prior knowledge of programming skills to model, manage rules, infer, and specify actions. The system adapts the SOA style architecture and has a web browser-based interface similar to a SMART SOA system. The results of the study indicate that the system is easy to use; however, the performance of the reasoning degrades with the increase in the number of models and the complexity of the rules. Likewise, [246] provides a pioneering OPEN framework. The OPEN framework is based on ontology for rapid prototyping, sharing and personalisation of the system for the cooperative use of the developers, and non-expert users.

Several other related works exist in the literature. For example, the assistive system [247] enables remote assistance and monitoring between the hospital and the clients' SH environment. Another study [248] proposed an SOA-style architecture involving a mobile device and a web service to detect objects in real-time by using image analysis techniques and augmenting the assistance on the user's tablet; here, a DD approach is employed through which images in the database are analysed. Meanwhile, the work in [249] adapts the KD approach to propose a multi-tier architecture for an autonomic Ambient Intelligent system. The system exploits ontology modelling techniques and logical rules [Java Expert System Shell (Jess)] to describe the environment formally as well as to infer and reason the activity. In addition, [250] fused the DD and KD techniques to achieve unusual behaviour recognition with the help of Decision Support Systems (DSS) and the ontology modelling technique for activity inferencing. The system provides a natural interaction (i.e., speech and gesture) within the smart environment and everything is controlled by the centralised server.

8.2.2. Relational and Graph-based Data Storage

Several mature relational databases (i.e., InfluxDB, MySQL, OracleDB and PipelineDB[251]) and graph-based database (Apache Jena Fueski Server [136], Neo4j, OrientDB, and ArangoDB) are available in the market. The relational database store data in a tabular manner (key-value pair), whereas graph-based database store values with a new relationship to form a set of triplets. One of the relational databases that highly optimised for real-time time-series analysis on continues data stream is PipelineDB. PipelineDB has many features such as sliding window queries, continuous aggregation and joining stream tables. Work in [252] compares query performances with mature MySQL relational database and popular Neo4j graph-based database. In addition, query performance based on RESTfull API and WebSocket connection are compared. The results indicate that Neo4j is a faster back-end database compared to MySQL and WebSocket connection performs better than RESTfull API. These results are also supported by [253] when two graph-based databases (Neo4j and OrientDB) and one relational database (MySQL) were compared with syntactic with big multimedia sensor data. Amongst the two relational databases, Neo4j outperforms OrientDB. However, the authors in [253], highlight that despite Neo4j being the leading graph-based database, it may not be suitable for the big data world. More specifically, Neo4j adapts master-slave approach which can scale vertically compared to OrientDB, which adapts a master-master approach to scale horizontally.

8.2.3. Human-Computer Interface for AAL Systems

Smartphones have become more, ubiquitous and have been integrated part of the modern lifestyle. Smartphones are continuously becoming more powerful with a diverse number of embedded sensors. In the future, these devices can be used for better contextual data collection as well as better usability of the system. In addition, delegating resource-intensive tasks to cloud-based service approaches can further increase the capabilities of smartphones and open up endless possibilities, such as Mobile Cloud Computing (MCC) [254], Cloud-based Mobile

Augmentation (CMA) [255], and Image Recognition processing (i.e., mobile landmark recognition systems) [256].

The old browser-based applications in previous SMART system implementations make a system less accessible to its users. For instance, the patients and caregivers would need to carry a laptop, tablet, or other browser-based devices to interact with the web service in order to receive real-time assistance. Furthermore, a browser-based application may not be able to utilise all services available on the device, whereas built-in hardware components, such as a heart rate sensor, can be used to detect/monitor the users' inactivity. In addition, hardware devices can be attached to mobile devices using wired or wireless communication protocols, such as Bluetooth, NFC, and Infrared. This capability allows limitless possibilities to collect diverse types of contextual data about the user. More importantly, the mobile application supports patients, caregivers and other stakeholders of the system (e.g., a patient's family members and relatives) on the move.

The main benefits of using the mobile device can be numerous. For example, it would not only allow the inhabitant to have a better HCI but also enable the utilisation of embedded sensors within the device or the attachment of external devices using wireless connectivity (i.e., Bluetooth). Such devices, such as Smartwatch and Shimmer [257] sensing devices can be used to obtain additional contextual information about the inhabitant to increase AR accuracy, which in turn, can lead to the provision of adequate assistance.

However, despite the advantages of using the smartphone application, providing every patient in the care home with a smartphone may not be financially feasible and getting the elderly to use it can pose further challenges. Therefore, providing efficient and natural HCI methods for an elderly can reduce those problems to a degree. For instance, the recent introduction of devices, such as Amazon Echo [122] provides voice-based interaction to the system and the ability to interconnect with a smartphone and other smart devices using SmartThings [123], can be advantageous.

This chapter makes three key contributions based on the findings from the literature review. Firstly, a microservice system architecture is proposed to delegate tasks such as sensor data collection from multiple data source, activity reasoning process and application scenarios to individual web services. This structuring of the AAL system will optimise the cloud computing configuration to achieve higher efficiency in performance, maintainability scalability and availability of the system. The second contribution is the development of Smart Lab based with real-time multimodal sensors from multiple communication protocols deployed in an ambient environment and embedded within the everyday objects. Finally, ODI guideline is followed to enable other researchers and AAL platforms developers, to retrieve, parse, visualise, analyse and understand the dataset more efficiently. Therefore, web interface and mobile application are developed and described. The usage of the mobile phone's sensor capabilities can also play a role in supporting additional application scenarios for the inhabitant and improving the system's usability.

8.3. Microservices-based System Architecture (MSA) for AAL System

A microservices-based system architecture (MSA) is proposed, which enables several tasks to be delegated between five essential web services, as illustrated in Figure 8.3. The SmartWeb API web service's role is to fulfil the requests made by external client devices. The SmartWeb API web service liaises with four primary internal web services to route the client's requests to relevant web service. These four internal web services are: Application API, Service API, Sensing Platform API and Big Data Storage API. The responsibilities of each internal web services are discussed in the following sections.

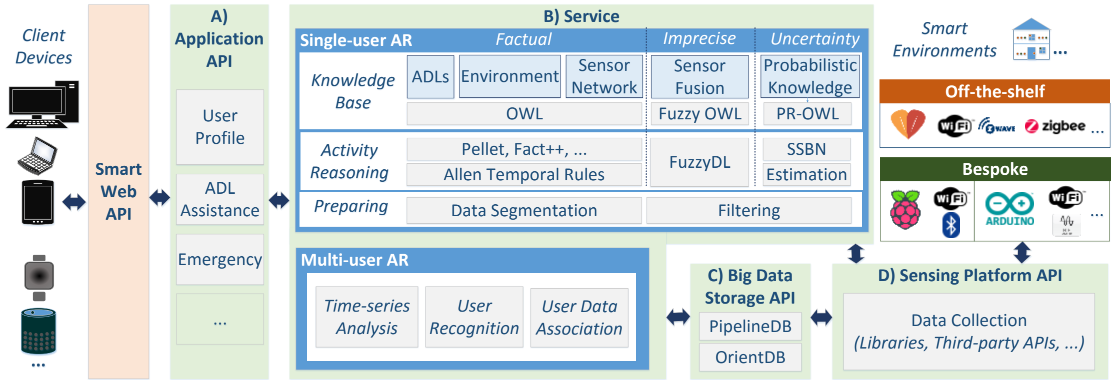


Figure 8.3. Microservices System Architecture (MSA) for Multi-user AR in a Shared Living Smart Environment

8.3.1. Application API

The role of the application API web service is to support five key features with the support of other internal web services. These five features are user profile management, intelligent notification services, ADL assistance provisioning and reminder services. The user profile management feature enables details such as resident's personal details, ADL preferences, relevant medication and doctor's appointment records to be stored using Big Data Storage API (more details in section 8.3.3). Other users such as carers have further details on ADL assistance provided to the patients, helping residents with their medicine intake and accompanying them to doctor's appointments. The key role of the ADL assistance tasks is to automatically detect emergency notification on a situation such as a fall or monitoring resident's ADLs to provide just-in-time assistance. The intelligent notification feature enables prompting both carer and residents in a non-intrusive way on their smartphones or SH devices based on their location (i.e., by raising the alarm, flashing LEDs or announcements with smart speakers). The reminder feature enables both carers and residents to manage their daily schedule and receive notification and track their activities over time. The scheduling for daily activities can include such as medicine dose preparation/ordering, and appointments for doctors and food shopping.

8.3.2. Service API

Service API web service is the core component of the AAL system. The ADL assistance feature in application API relies on the Service API to analyse the sensing data received from the Sensing Platform API web service. Service API perform three main tasks, semantical sensor data segmentation, filtering data and human activity recognition with the reasoning engine.

The semantical segmentation [153] approach, detailed in CHAPTER 3, is leveraged to separate and group sensor observation based on their relationship with ADL descriptions specified in the knowledge model. The knowledge engineering technique is being leveraged to describe generic everyday objects within the living environment and ADLs into the ontological model. An incremental pellet reasoner is used to derive facts that are not expressed explicitly in the ontological model; this process is known as terminology-box reasoning (T-box). The user-specific preferences are also described as instances of a specific ADL class and stored using Big Data Storage API. This process is known as assertion-box reasoning (A-Box). Both generic and user-specific preferences knowledge is utilised to segment each sensor observation into a relevant set of ADL queues.

The sensor observation data, such as accelerometer and gyroscope are prone to drift in their reading over time. Hence, filtering and smoothing techniques such as complimentary and Kalman filter are required before performing activity recognition algorithms. The filtered observation values for a set of segmented sensors for a given ADL are evaluated using time series analysis with sliding windowing process.

The filtered sensor data is used to perform activity recognition at multi granularity ADL level: coarse and fine-grain. The coarse-grain ADL level mainly considers criteria under which a given ADL must be fulfilled. These criteria are location, time interval and key objects. These set of criteria for a given ADL is stored in the ontological model and can be queried using SPARQL Protocol and RDF query language (SPARQL) or description logic (DL) query approach.

The motivation of fine-grained level activity detection is to detect and verify if the intended interactions with the object have been conducted or not. For instance, "*pouring hot water*" from a *kettle* into the *cup* when making a *tea*. A few things can go wrong when performing this action. For example, the kettle might not have been turned on or breakdown; hence, the kettle water is cold, or the kettle and cup can be dropped or spilt in the process of pouring. Therefore, kettle and cup are required to host multiple object sensors to monitor varying attributes such as the liquid level, temperature and 3-dimensional position sensor (i.e., an inertial measuring unit (IMU) containing accelerometer and gyroscope sensors). Hence, a sensor fusion technique is required that combine the states of the sensing attributes to determine "pouring hot water" action is complete. Moreover, the measurements nature of sensors such as temperature and liquid level output imprecise value that is subjected to interpretation. The limitation of the ontological model is that DL based formal theory defines the relationship between object and subject to be 1 or 0; also known as crisp sets. Therefore, fuzzy ontology modelling and fuzzyDL based reasoning approach are developed to define imprecise sensor value as gradient values between 0 and 1.

The fuzzy ontology is formally based on fuzzy set theory. There are three key steps in developing a fuzzy knowledge base, fuzzification, rules and defuzzification. In the fuzzification step, the fuzzy concepts are defined using fuzzy membership functions. For instance, kettle temperature "*very hot*" can be defined using right shoulder membership function *(a=50, b>=70, min=-10 and max = 150)*. Similarly, a liquid level "*medium*" (measured in picofarad) can be defined using triangular membership function *(a=18, b=25, c=35, min=0, max=50)*. The fuzzy rule is then created using antecedent (IF) and consequent (THEN) statements. For example, the antecedent can be represented using *(define-concept rule1_antecedent (g-and (some hasLiquidLevel kettleLiquidMedium*) (some *hasTemepratureLevel kettleTempVeryHot*)) and consequent using *(define-concept rule2_consequent (g-and (some hasPouredStatus success))*. The two statements can be combined with the third statement *(define-concept hasPouredRule (l-implies rule1_antecedent and rule2_consequent)).* The final step involves performing defuzzification using methods such as Centroid Of Area (COA), Bisector Of Area (BOA), and Mean Of Maximum (MOM) [166]. The defuzzification method function requires instances containing multiple sensors attributes to output the result of poured status to be 1 (success) otherwise 0 (failure).

The sensing environment is prone to be affected by many factors which can cause uncertainty with the data received and the confidence of AR results. Some of the factors creating uncertainties are a human error based, sensor failure, low battery and interferences in environmental factors causing data packet loss or corruption during network transmission. Therefore, probabilistic reasoning ontology (PR-OWL) based approach is proposed further to extend the expressive capabilities of OWL and Fuzzy OWL. Details of the probabilistic, fuzzy and combined framework to support imprecise and uncertainty reasoning for AR is provided in CHAPTER 4, CHAPTER 5, and CHAPTER 6, respectively.

8.3.3. Sensing Platform API

The Sensing Platform API web service centrally collect smart environment data from multiple sources. The two main components of Sensing Platform API web service are software and hardware. Both software and hardware architecture components are proposed to be developed using bespoke, off-the-shelf and open-source libraries and devices to support the experiments from previous chapters.

The central role of the software component is to provide real-time sensor events to external clients and internal web services via Smart Web API. The heterogeneous wired and wireless sensors network deployed in the smart environment is modelled by integrating Semantic Sensor Network (SSN) vocabulary into the ADL ontological modelling process (section 6.3.1.3 elaborated on the use of SSN classes). Entities in the smart environments can have one or more sensors attached or embedded within them and have a separate module/device/platform to process and transmit the data to Sensing Platform API. The purpose of embedding multiple types of sensors into an everyday object is to allow researchers to test the algorithms developed in Service API for performing AR and detecting individual user's intended action with the object at satisfactory completion threshold for a given ADLs at higher accuracy. Further details on the rationale, method and techniques used to perform fine-grained action level AR with multimodal sensing attribute is conveyed in CHAPTER 4. Moreover, the data generated from the sensing environment is stored using Big Data Storage API (more details in section 8.3.4).

The selection and configuration of hardware devices are depended on multiple factors such as sensing attributes required, types of sensors needed and cross-manufactures hardware compatibility. Wide range of sensing methods is available with diverse communication protocol as discussed in Section 2.3. One of the main challenges of developing a smart environment is first to find the type of sensors required to the application from the platform compatible manufacturer and second integrating manufacturer platform with your system using their third-party APIs. However, it is likely that not all the sensor types will be available with compatible devices. Hence, bespoke microcontroller-based sensing is proposed to give more control in the sensing behaviour required for the system.

A sensing hardware architecture proposed in this section and depicted in Figure 8.4 based on the evaluation requirements for single and multi-user AR studies presented in this thesis. The setup consists of ambient and dense sensors (indicated by a blue and red bar) where multiple communication protocols and microcontrollers (acting as aggregators) are leveraged. For ambient sensing, off-the-shelf door/window, motion, and multi-sensors (i.e., temperature, humidity, luminosity, vibration, motion and door/window magnetic) sensor are connected to

Securifi almond router via Z-wave and ZigBee communication protocols. This almond router exposes the sensor data to the client devices that are connected to the home network, i.e. the web service via a web socket connection. The dense sensing is performed using Arduino based microcontrollers which are either directly connected to the web service using the universal serial bus (USB) or over a WIFI connection. The key sensors integrated within these microcontrollers are capacitive liquid level (FDC1004) and touch (TTP223B) sensor. To collect object positions attributes, i.e., accelerometer and gyroscope, BLE based TI SensorTags (CC2650) are used which contain additional sensors on board. Currently, up to 5 TI SensorTags can simultaneously interact with one Raspberry PI and on-demand expose the data to the clients over web socket connection; in this case, the web service.

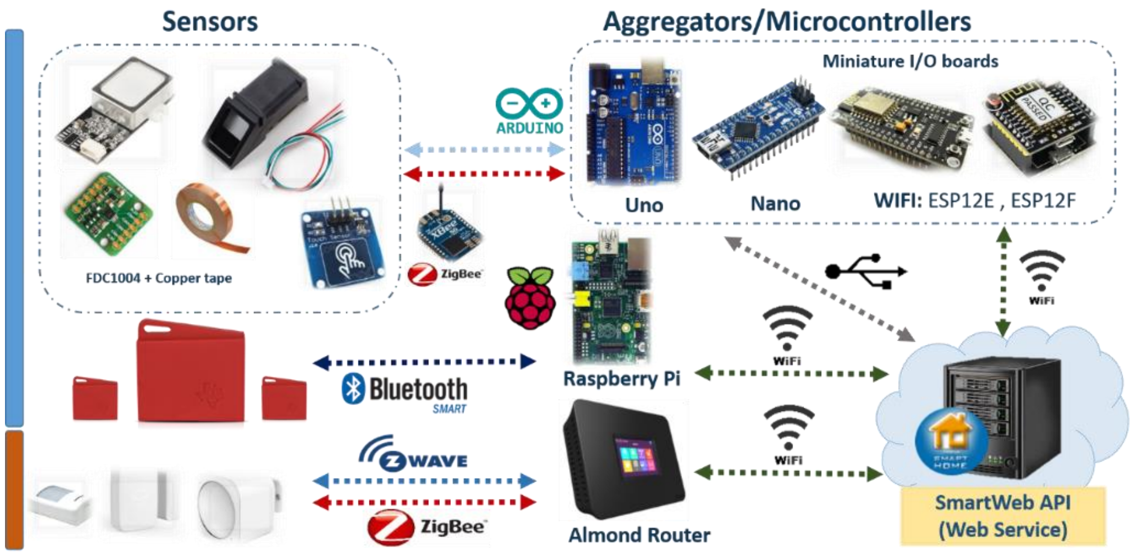


Figure 8.4. A Sensor-based Hardware Architecture with Multimodal and Diverse Communication Protocol.

8.3.4. Big Data Storage API

The role of the Big Data Storage API is to provide an interface for other three internal web services to efficiently read, write and update a large amount of data in real-time. To meet the demand for data segmentation to process continuous sensor data stream, event-based relational PipelineDB is proposed. However, to store other semantical metadata such as sensors configuration, AR results, and user profiles, OrientDB graph-based hybrid approach is proposed. OrientDB provides high scalability for big data with master-master database architecture, speed and the ability to store multi-modal data.

8.4. Implementation Details

The proposed MSA based SMART system has been re-engineered to perform AR within both simulated and real environments. MSA follows a client-server model where a collection of web services works together to fulfil the requests/responds to the multiple clients simultaneously. Implementation details of client web-interface and Android mobile application is provided in section 8.4.1. In addition, implementation details of web services and dataflow between client and smart environment are provided in section 8.4.2.1. A smart lab environment was created to create a kitchen environment with real-time binary and multimodal sensors in two stages; more details in section 8.4.3.

To analyse the data collected from the smart environment, the system can currently conduct data segmentation in real-time and simulated environments and AR with actions at a fine-grained level using simulated/pre-collected data. In addition, several features were added to mobile and web interfaces such as to visualise real-time/pre-recorded multimodal data, add user preferences, manage medication dose, sensor management dashboard support and dataset conversion tool from JavaScript Object Notation (JSON) file to Extensible Markup Language (XML) file with HomeML schema (ODI framework). Details for some of the semantical data segmentation and fine-grained AR are already provided in respective chapters. However, section 8.4.4 provides details for knowledge and reasoning for additional features stated above.

The proposed MSA was initially developed using SOA with a single machine and single multi-layered web service[153], [157]. The initial prototype was successfully developed and evaluated to perform semantical segmentation and SPAQL based inferencing with real-time binary sensing environment. However, to perform single-user AR tasks with fuzzy data and uncertainty reasoning with multimodal sensors environment, a single machine with limited cores and threads could not conduct the tasks in a reasonable time or maintain the growing system overtime. Hence, MSA was proposed to further delegate AR tasks over multiple computers/microservices and create additional computational resources. The development of MSA with a real-time multimodal environment is complete and detailed in sections 8.4.3 and 8.4.3. However, more effort is required to complete the integration of single-user AR framework and multi-user AR proposed in CHAPTER 6 and CHAPTER 7, respectively.

8.4.1. Feature Rich Mobile and Web Interface

An adaptive web-browser interface and mobile application based on Android operating system (OS) were developed to make asynchronous Hypertext Transfer Protocol (HTTP) request or establish a web socket connection to retrieve live sensor data and AR results from the Smart Web API web service. Android OS based mobile application was developed due the availability, popularity, and extensive community support for Android applications developers. In general, the overall system client-server architecture platform is flexible to support any other operating systems using API and HTTP communication protocol. In addition, new features can be flexibly added to further assist the inhabitant in living independently or in the care home.

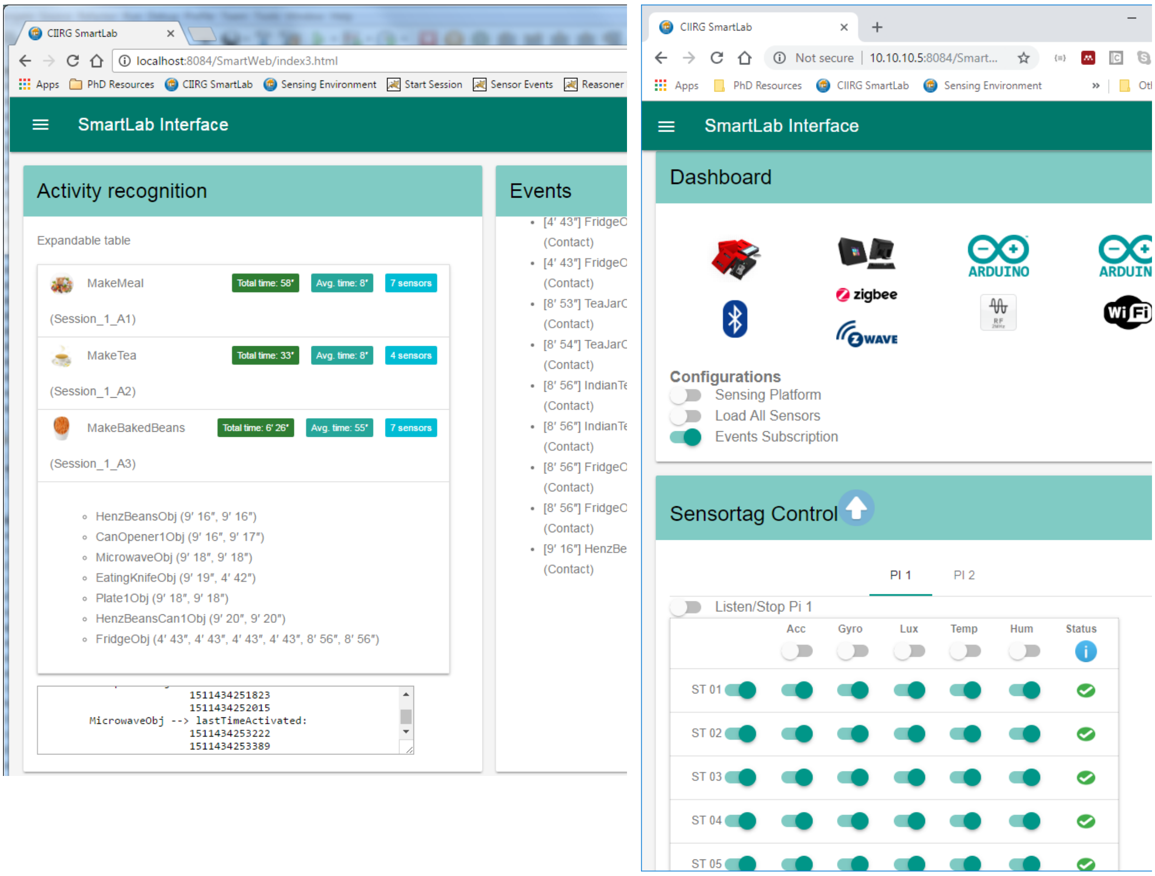


Figure 8.5. Web interface: (left) Activity recognition page containing sensor events and results, (right) sensor configuration dashboard

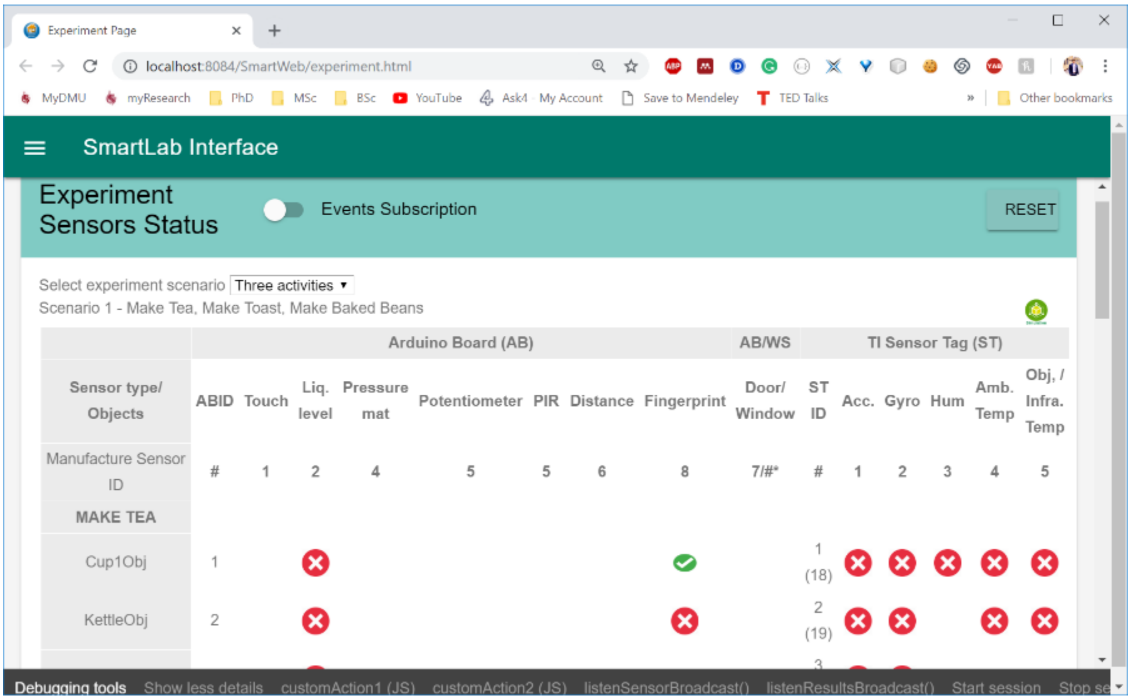


Figure 8.6. Experiment Tool to Check Sensor Status based on Sensing Platform attached to Objects

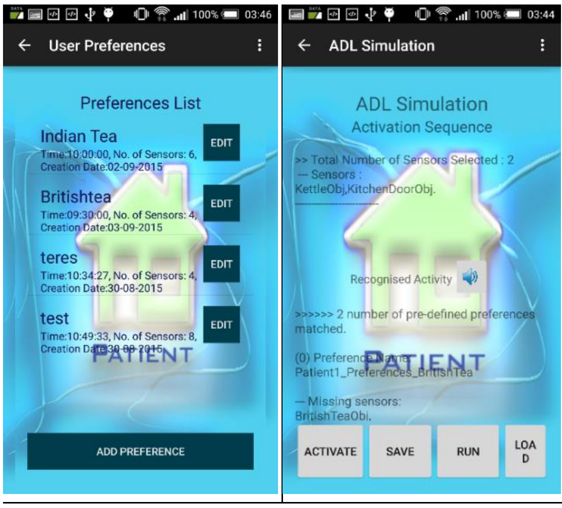


Figure 8.7. Management of user preferences (left) and ADL simulation interface (right)

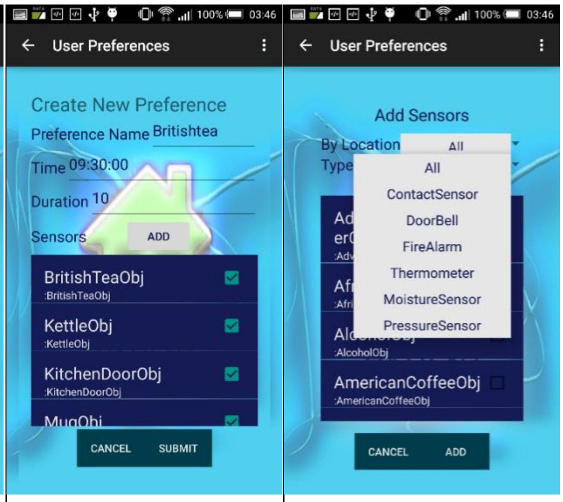


Figure 8.8. Creating a new user preference interface (left) and filtering sensors list by location (right).

A fragment of the web-browser interface is provided in Figure 8.5 and Figure 8.6. Figure 8.5 shows a screenshot of the web-browser based interface to allow the user to interactively view ongoing activities and live events logs on the left-hand side. The right screenshot in Figure 8.5 presents a dashboard to configure four types of platforms and sensors associated with it. These four types of platforms are TI sensor tag, Securifi Almond, Arduino microcontrollers adapting radio frequency protocol, and Arduino microcontrollers with WIFI shield. The web-browser interact with Smart Web API to retrieve smart environment configuration. The Smart Web API liaise with internal web services such as Sensing Platform API for interacting with sensing environment and view the live status of the four types of platforms. In particular, Sensing Platform API maintains web socket connection with the Raspberry PI to give commands to individual TI Sensor Tags. Raspberry PI acts as an aggregator to communicate with TI Sensor Tag using short-range Bluetooth communication and sensing platform with an internet connection. Similarly, Figure 8.6 shows a screenshot of the experiment tool developed to check the status of the sensors under different activity test case scenarios. In addition, the interface allows user to subscribe to live sensor events or load previously recorded sensor events log and update the sensor status for the experiment using the green and yellow button above the scenario table located on the right-hand side.

A fragment of an Android mobile interface is provided in Figure 8.7 and Figure 8.8, in addition to the live sensor log and AR results interfaces in previous chapters, i.e., 2.8. Figure 8.7 show user interfaces to allow users to manage their ADL preferences on the left and conduct ADL simulations on the right. Android application uses a simple model-view-controller (MVC) design pattern to separate the classes logically. The model package contains all of the domain models that are used to map the data communicating with the web service. The view package can be composed of all the classes that are being used to display views on the screens, i.e., activity classes, fragment classes, and dialogue classes. Depending on the user types, the view package may have further sub-packages to separate all the views. The controller package may consist of all the classes that trigger requests to the server with the help of the utility classes, mainly view listeners and adapters. Finally, the utility package holds all the support classes, such as HTTP async requester classes, data parsing classes, data dictionary classes, and date format utility.

8.4.2. Web Services

The web services adapt lightweight Representational State Transfer (REST) based software communication architecture style to reduce the payload of the request/respond packets and increase efficiency. However, other communications protocols such as Simple Object Access Protocol (SOAP) with defined request requirements for clients and rich built-in features such as security. The REST-based web service has been identified to be better suited for the AAL system based on the following reasons. The REST-based protocol is lightweight in nature and is easy to use and implement compared with the SOAP web service. The SOAP-based protocol supports richer functionalities but incurs communication overhead [117], [258]. In addition, it poses restrictions in terms of flexibility, explicit functional parameter requirements, and the data format that it can produce and consume. In comparison, the JAX-RS library [193] in the REST-based service does not require function parameter definitions or publication of their service, i.e., with universal description, discovery, and integration (UDDI). Another main feature of the REST-based service is that it enables clients to consume and produce data in a variety of data formats, such as XML, JSON, HTML and encoded text. Thus, making the system more interoperable compared with others and gives it the ability to support low-powered devices, thus reducing their limited energy consumption resulting from its light weight nature.

MSA approach essentially follows a client-server pattern, in resolving some of the technical challenges mentioned above in building an assistive system using the SH environment. For instance, a Web Service as a service provider and a Mobile application as a client can work well together to bridge the communication gaps between the SH environments and mobile device as well as to, make the system more flexible in terms of scalability, performance, and platform independence. Furthermore, the web service can take advantage of cloud computing technology to increase the ability to perform complex reasoning or computation tasks effortlessly.

One of the main requirements for the web service is to capture and expose all the sensor data and activity inferencing results to the client devices upon user interactions with the environment. For this, sensing platform API broadcast the real-time sensor data to the clients using the Server-Sent Events (SSE) [259] mechanism instead of a bidirectional WebSockets or pooling method. One of the main reasons for this is to reduce connection overhead. Although SEE is a bi-directional protocol, other standard requests can still be made by a client outside their SSE connection asynchronously. Another requirement of a web service is to capture and process sensor data that are communicated to the server in various media formats depending on the device vendor. In this proposal, the web service currently supports Almond+ router WebSocket connection, XBee coordinator connected via comport, and another Arduino-based sensor collection using standard comports (see sections 2.7.6 for more details).

The Jena Fuseki server has been used as it supports the Java programming language and works well together with the Apache Jena API [260] used to flexibly change reasoners and perform SPARQL queries on the graph models. Furthermore, the Jena Fuseki server supports various development tools, such as command-line execution of the data (ARQ), and user-friendly web-based interface to compose, execute queries and manage multiple datasets. However, to achieve a distributed collection of data for higher scalability, reuse, and performance; however, other triplestores discussed in section 8.3.4 can be incorporated to support big data requirements and optimised stream processing.

8.4.2.1. Dataflow Between the Client Device, Web Service, and Apache Fuseki Server

Smart Web API web service is central to the client interfaces (web-browser/android application) and Apache Fuseki Server. The Android application makes standard HTTP requests (i.e., GET, PUT, POST, and DELETE) to the web service to perform several tasks, such as CRUD operations, inferencing, reasoning, and other complex application-based logics. All the RDF data and ontologies are stored in the Apache Fuseki Server as a graph. Therefore, the data are retrieved and manipulated by the web service using SPARQL query language with the support of Apache Jena library and the standard HTTP protocol. However, the real-time sensing data are exposed to the clients using a half-duplex, listener-subscription mechanism (i.e., Server-sent events (SSE) [259]) in comparison to full-duplex WebSocket. One of the critical reasons for this decision is so that the process-intensive tasks of inferencing and reasoning are performed independently of the real-time event logging process.

The web service broadcasts two SSE methods to the clients: one for broadcasting real-time sensor events and another with inferencing results for the clients with a session token. This sequence of events between the client device and the key components in the web service is illustrated in Figure 8.9 below. As can be seen, the client Android application can listen to the sensor events in the background asynchronously by making an SSE call to "*EventBroadcaster*" function in the *SensorsCall* class located in "*SmartWebServiceAPI*" (A). To receive client-specific inferencing results, the client must obtain the session identity from the "*ReasonerCall*" first (B). The "*ReasonerCall*" is responsible for the task of listening to the sensor events from the given time, performing inferencing and then broadcast the result using "*ResultsBroadcaster*" function (B.1). Once the client receives the session token, a request can be made to "*ResultsBroadcaster*" after which the task of listening to the inferencing results associated with their session identity is initiated. Meanwhile, the client device is responsible for closing the session (C) and, if required, storing the session data separately.

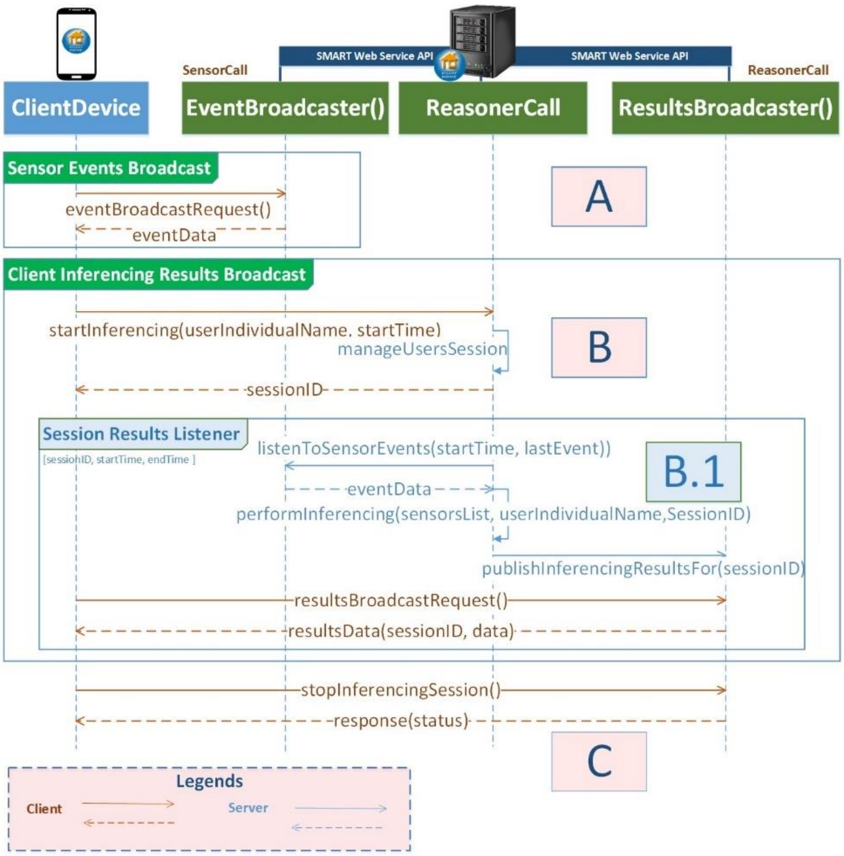


Figure 8.9. Server-sent event (SSE) mechanism for real-time message flow of sensing and inferencing results between client and web service

The web service performs a query and an update request in three simple steps: (1) building SPQARL query/update string, (2) using Jena classes/standard HTTP post methods to execute the request, and (3) parsing the responses. The pseudocode, shown in Figure 8.10, performs a simple SPARQL query on the local Fuseki server endpoint and parses the result using the *ResultSet* and *QuerySolution* method. The standard HTTP post request can be made to perform SPARQL update using the *HttpPost*, *HttpClient*, and *HttpResponse* classes. However, the request content type is set to "*application/sparql-update*", and a static variable already defined in the Jena's WebContent class ("*WebContent.contentTypeSPARQLUpdate*") can be used.

```
// 1) Building Query String (Inc. all the prefixes for the vocabularies)
String queryStr = DataDictionary.PREFIX_DEFAULT
            + “SELECT * “
            + “WHERE { ?class rdfs:subClassOf :Sensor. }”
            + “LIMIT 100”;

//2) Using Jena query to execute the SPARQL query
Query query = QueryFactory.create(queryStr);
QueryExecution qexec = QueryExecutionFactory.sparqlService(“http://localhost:3030/ds/query”, query)
ResultSet rs = qexec.execSelect();

//3) Iterate through the result using column names, in this case ‘class’
while (rs.hasNext()) {
   QuerySolution soln = rs.nextSolution();
   //populate all the columns
   for (int i = 0; i < columnNames.size(); i++) {
         RDFNode rnode = soln.get("?" + columnNames.get(i));
   }
}
```

Figure 8.10. Pseudocode for executing a SPARQL query on the server endpoint using Jena API

Next, the Android application makes the requests to the web service using the standard HTTP protocols (*HttpGet, HttpPost, and HttpPut, HttpDelete*), only in a JSON format; hence, the request headers must be set appropriately. The Android application parses the JSON data, and by using the "*org.codehaus.jackson.map.ObjectMapper*” class, the data can be automatically remapped into their respective class instances.

8.4.2.2. Data Collection from SH Environment and Storage

As discussed in previous sections, a diverse number of sensors and communication protocols are currently available in the market. The proposed architecture currently uses the Securifi Almond+ router to perform ambient sensing, Arduino boards for dense sensing, and Amazon Echo for voice interaction (see section 8.4.3 for configuration details). The Securifi Almond+ router is used as a main IoT (Internet-of-Things) hub because of its WiFi, ZigBee, and Z-Wave protocol capabilities. Other hubs supporting similar protocols are also available, such as Libelium Waspmote [261], SmartThing Hub, and VeraLite. However, further investigation may be required to obtain real-time data from these hubs. The popular Arduino boards and shield-based approach provides more exceptional capabilities and flexibility with which to perform sensing; however, additional steps are required to configure the individual components. Meanwhile, the Amazon Echo currently supports WiFI and Bluetooth communication protocols, thus allowing voice interaction capabilities with third party services.

In relation to overall system architecture, the “*Utility*” library consists of packages and classes through which to extract, store, and process the data from the sensing hardware devices. In particular, the “*Sensor Utils”* package contains sub-packages and classes that interact with third-party APIs and hardware libraries (i.e., “**.almond*” and “*.arduino*”). Some of the key Java libraries used are WebSocket API (for Almond+ router), XBee, and comPort (both for Arduino). Moreover, these classes are used by the parallel thread classes to log the events (“*EventLogThread”*), perform device management (“*DeviceManagementThread”*), and store the data in the triple-store (“*TDBStorageThread*”). Figure 8.11 illustrates the abovementioned utility library structure.

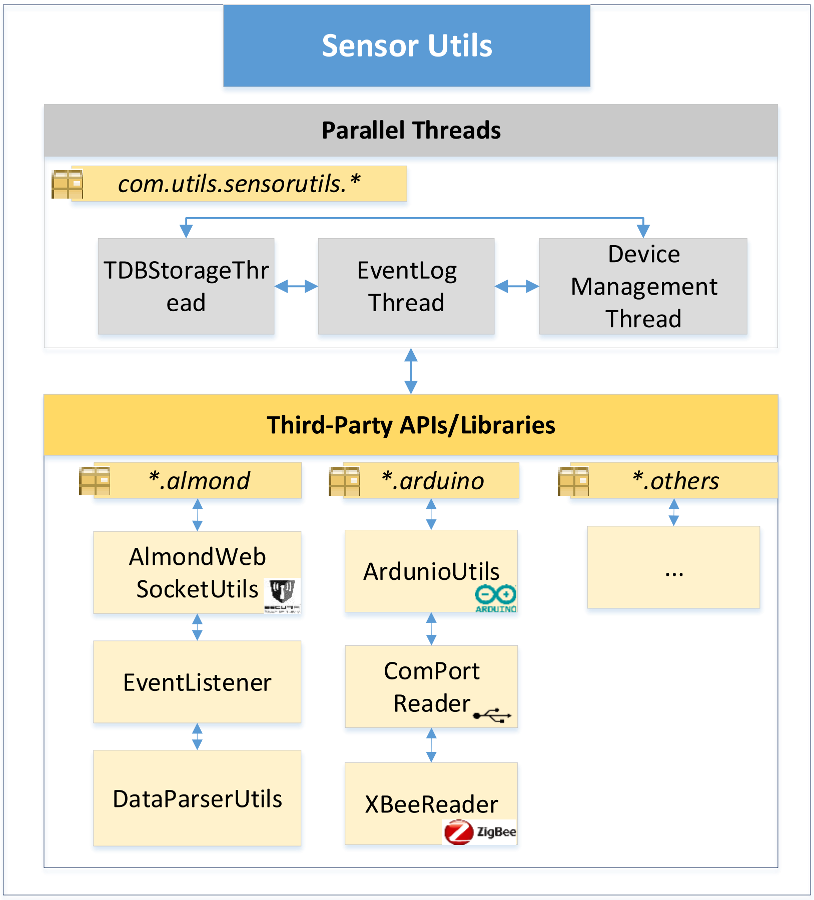


Figure 8.11. Software: Breakdown of the “*Sensor Utils*” package

The system implementation currently stores all the sensor events log, AR results, user profiles, and other data in the graph-based database, Apache Jena Fuseki Server, instead of hybrid use of PipelineDB and OrientDB database. The results are exposed to client devices via SmartWebAPI using RESTful communication protocol and JavaScript Object Notation (JSON) data format. More details of system architecture and hardware sensing configuration can be seen in our previous work [157], [158]. Moreover, Jena Fuseki supports popular Java programming language and it works well together with the Apache Tomcat server, and Jena API to perform SPARQL queries and reasoning with Pellet. In addition, the Jena Fuseki has a number of features such hosting database within a web application or externally, command-line tool (ARQ), a user-friendly web interface to manage graph data.

8.4.3. Multimodal Smart Environments

The smart lab environment was developed in two stages. In the first stage, mainly binary sensors were distributed in the ambient environment and embedded them into everyday objects. In contrary, multimodal sensors were added to ambient and embedded sensing environment in stage two. The main purpose was to collect additional user interaction information from a given object or environment for achieving higher accuracy in recognising user actions at a fine-grained level. Further details on reasoning and fusing multimodal sensor data is presented in CHAPTER 4. At each stage of the smart lab environment, different types of experiments were conducted to evaluate the proposed approaches as detailed in section 8.5.

Figure 8.12 presents the stage one hardware configuration diagram and deployed sensing environment in order to start collecting the raw data. Ambient sensing is performed using preconfigured sensors that are compatible with the IoT hub, i.e., door, motion, and multi-sensors. Embedded (or Dense) sensing is performed using bespoke configurations wherein Arduino Uno boards with XBee shields and modules are used to create a mesh network; see [262], [263] for more details. The main coordinator that receives data from the remote nodes is directly connected to the webserver using comport. However, other options are also available to send the data from the coordinator to servers, such as by using WIFI shields or Bluetooth. The remote nodes, which relate to various multimodal sensors and sends their statuses to the coordinator when an event is triggered. In addition, an Android mobile phone, Amazon Echo, and WeMo Sockets are also attached to the IoT router. The Android mobile phone is directly connected to the Amazon Echo via Bluetooth to output activity recognition results. In turn, the Amazon Echo can interact with the Almond+ router and with other popular sensing vendors. The WeMo Sockets and Amazon Echo can be easily integrated within the proposed mobile application using their APIs.

The stage two multimodal hardware configuration was proposed in Figure 8.4 and the distributed environment is presented in Figure 8.13. Each entity in the smart environment contains more than one sensor or sensing platforms to collect their change in status or attributes upon user interactions. The details of sensing devices, their sensing attributes and communication protocol used for stage two is described in section 8.3.3.

The smart lab environment was developed to create a kitchen environment with over 20 everyday objects used for making tea, toast and beans. It was equipped with 12 types of multimodal sensing data collected from 3 types of sensing platforms as detailed in Table 8.1. The room floor plan for stage two sensing environment is depicted in Figure 8.14 and distributed environment in Figure 8.15.

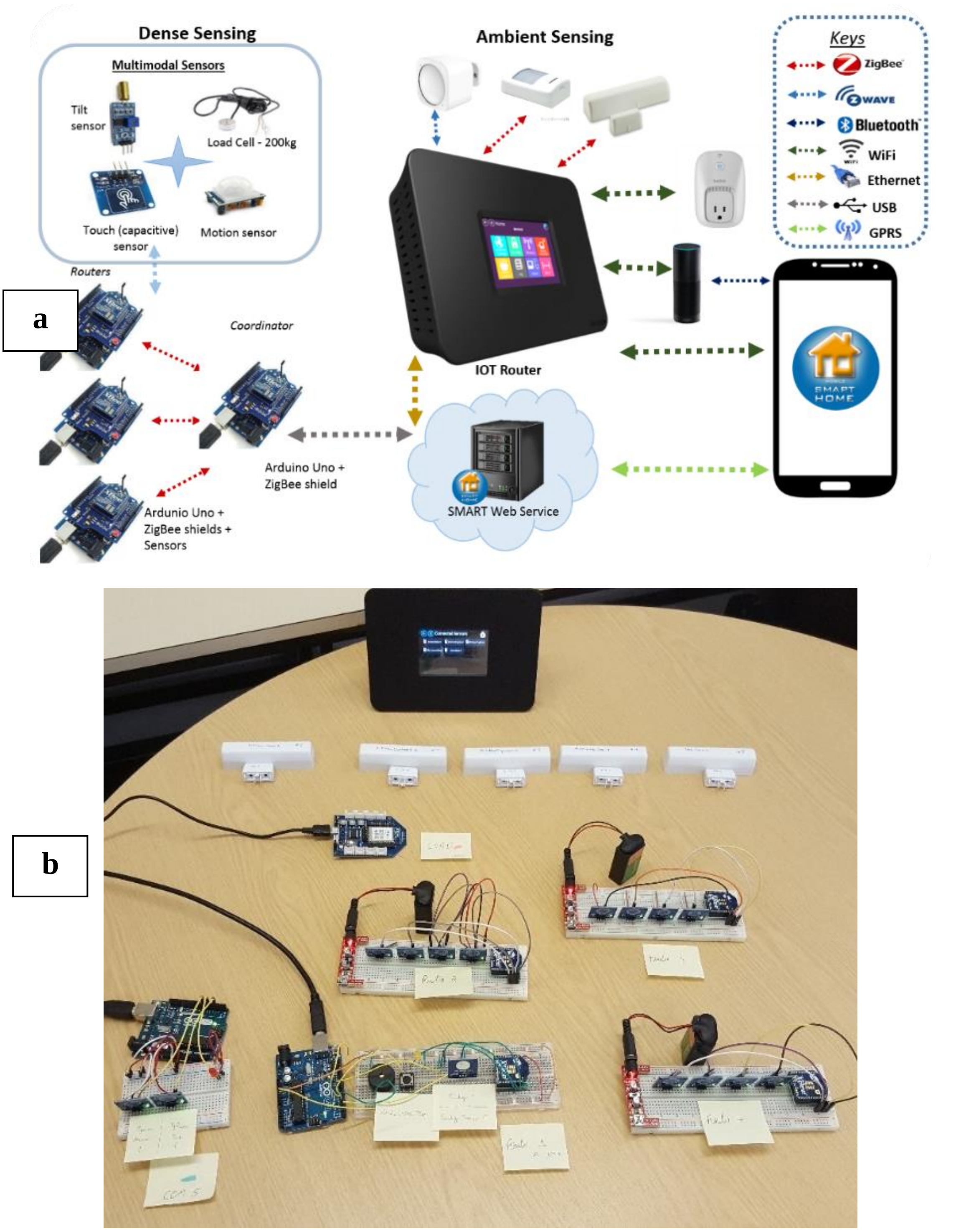


Figure 8.12. *Stage 1* – Smart Lab Hardware Setup: (a) Connectivity diagram of sensing devices and (b) hardware deployment experiment for simulating experiments.

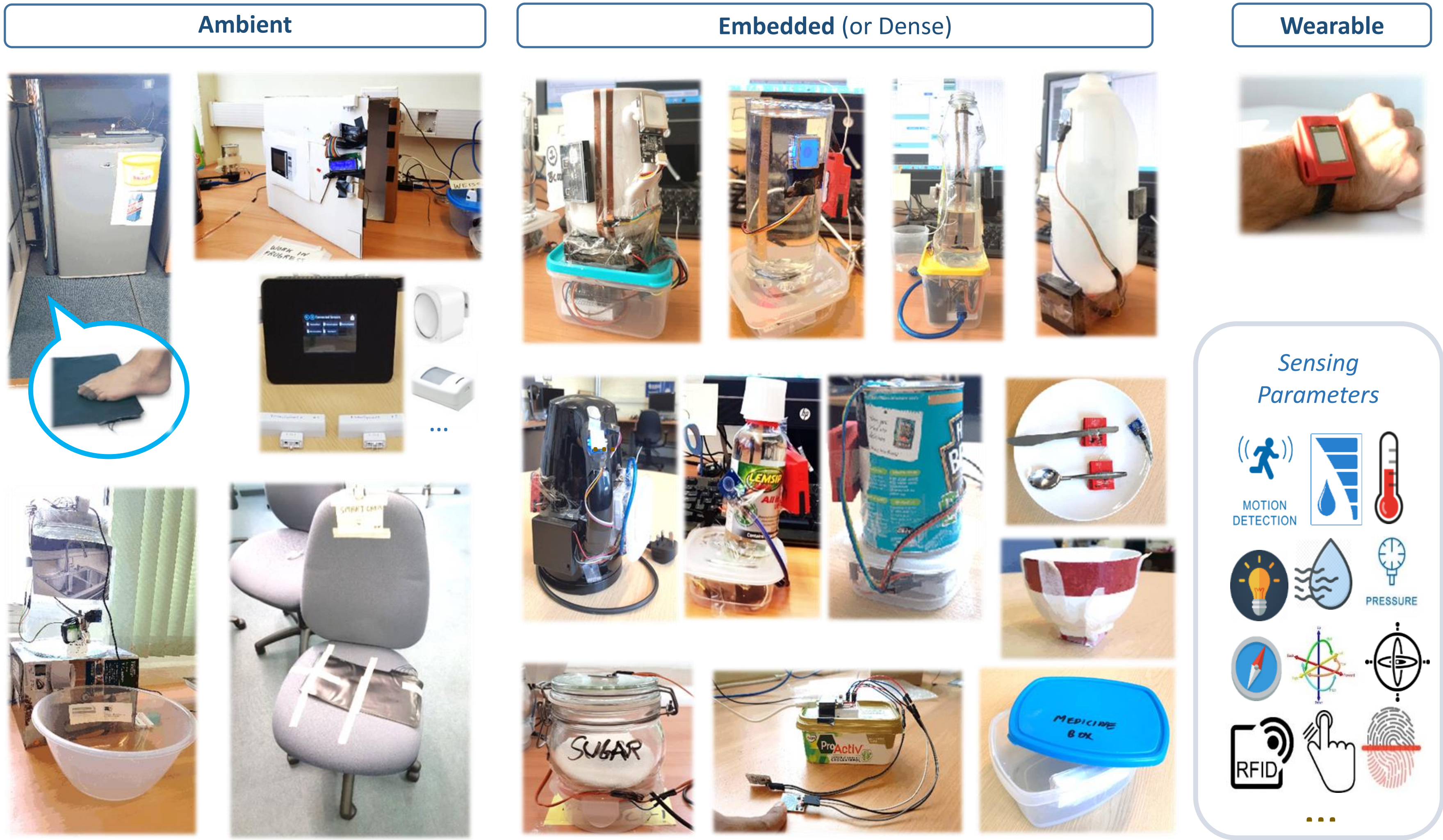


Figure 8.13. *Stage 2* - Deployment of smart kitchen environment with ambient, embedded and wearable sensing environment based on off-the-shelf and bespoke IoT-enabled microcontrollers with diverse communication protocol, i.e., Wi-Fi, Bluetooth, WebSocket, Zigbee and Z-wave.

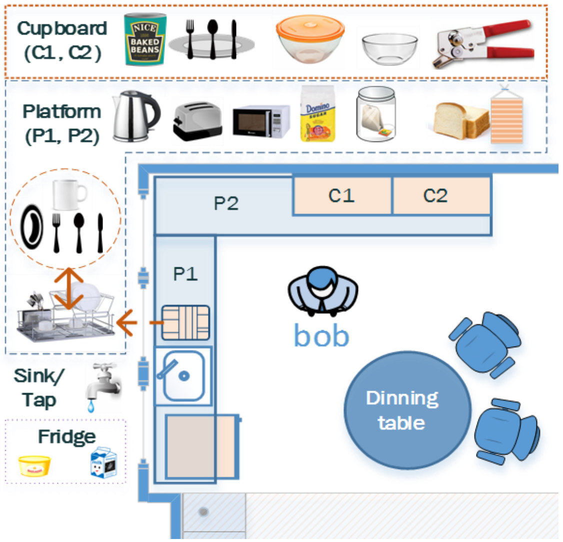


Figure 8.14. Smart kitchen floor plan with distributed objects.

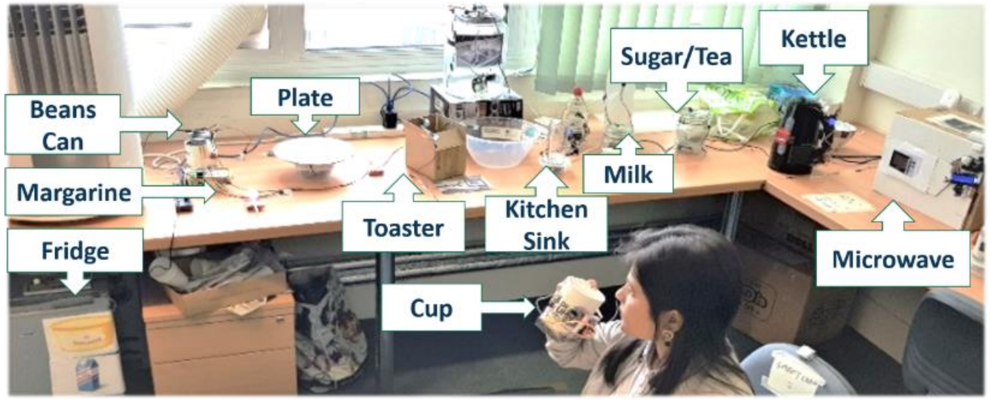


Figure 8.15. *Stage 2*: Smart Kitchen Environment for User Interaction Monitoring at Fine-grained Action Level.

Table 8.1. Everyday Object and their Corresponding Sensing Parameters and Platforms

Sensor type/ Activity & Objects	**Arduino**							**Sensor Tag**						**Securifi**
	AB ID	T	L	PM	PO	PIR	D	**ST ID**	A	G	H	AT	OT	D/W
TEA														
Cup	1	✓	✓					1	✓	✓	✓	✓	✓	
Kettle	2	✓	✓					2	✓	✓		✓	✓	
Dish Soap	3	✓	✓					3	✓	✓				
Water Tap	4	✓		✓	✓	✓	✓							
Tea/Jar	5	✓												
Sugar/Jar	6	✓												
Fridge	7	✓		✓		✓	✓							✓
Milk	8	✓	✓											
Spoon1								6	✓	✓				
TOAST														
Plate1	9	✓												
Bread Slice/Pack	10	✓												
Toaster	11	✓		✓		✓								
Fridge	7	✓		✓		✓	✓							✓
Margarine	12	✓												
Eating Knife								4	✓	✓				
BAKED BEANS														
Eating Knife								4	✓	✓				
Beans/Can	13	✓						5	✓	✓				
Spoon1								6	✓	✓				
Bowl								7	✓	✓		✓	✓	
Plate1	9	✓												

Note: {ABID: Arduino board ID, T: touch, L: liquid, PM: pressure mat, PO: potentiometer, PIR: passive infrared sensor, D: distance}, {STID: sensor tag ID, A: acc., G: gyro, H: hum., AT: amb. temp., obj. temp}, {Securifi almond router: D/W: door/window}.

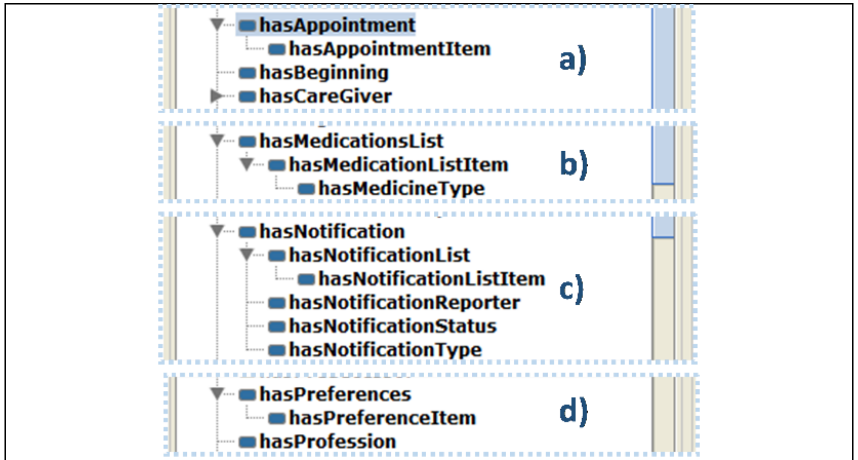


Figure 8.16. Layered object properties for bucket-based structure data

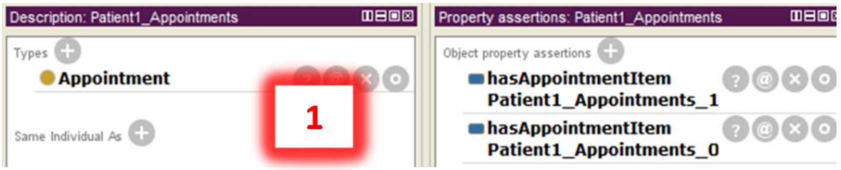
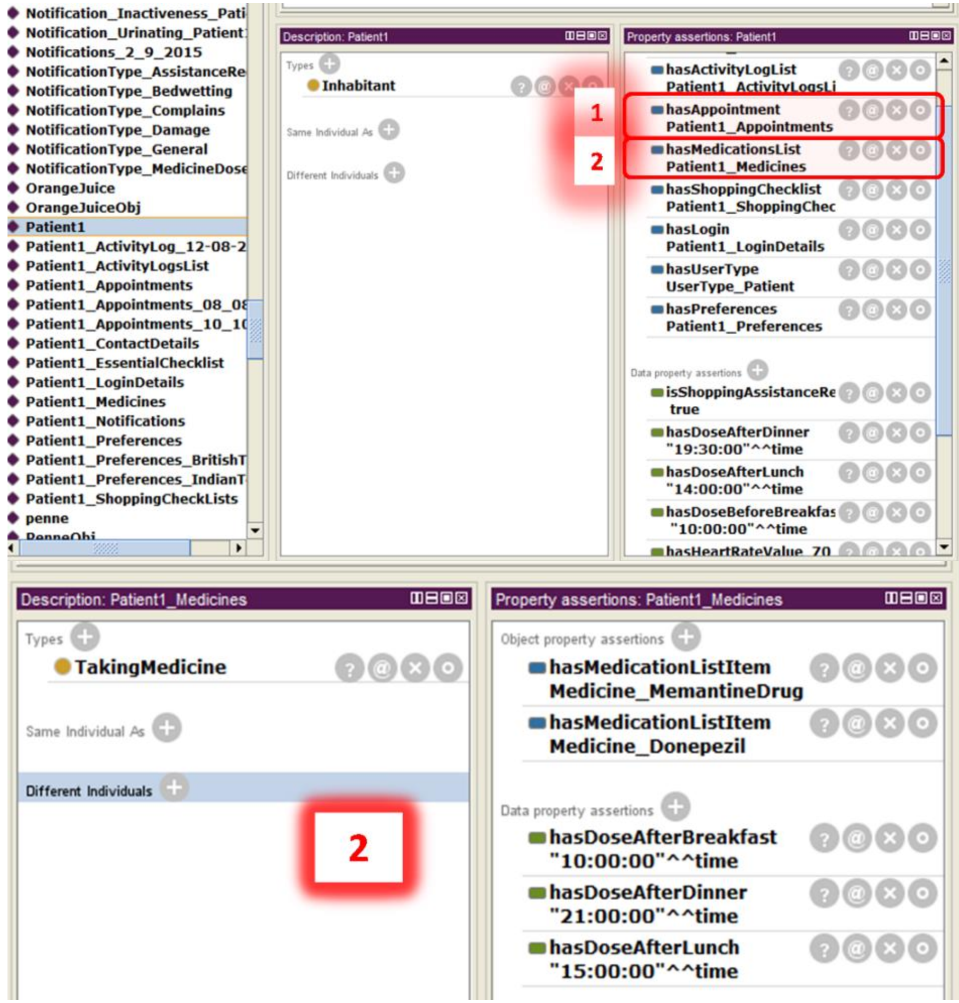



Figure 8.17. Bucket-based approach for data structuring to capture (1) appointments and (2) medications list for patient1 using *hasAppointment/hasAppointmentItem and hasMedication/hasMedicationListItem* object properties, respectively.

8.4.4. Activity Modelling and Reasoning

8.4.4.1. Assistive Feature Ontology Modelling

Extensive use of semantic-based ADL modelling and reasoning is provided in previous chapters to recognise single and multi-user activities. However, this section illustrates how additional assistive features such as (a) patient appointments records, (b) patient medication doses, (c) carer notifications services and (d) patient ADL preferences are also modelled using semantic knowledge modelling such as Protege[159]. Protege is one of the opensource ontological editing tool available to build a conceptual model at varying levels of abstraction, leading to the encapsulation of a particular set of knowledge.

Figure 8.16 and Figure 8.17 presents a bucket-based approach for modelling (a)-(d) features using classes, properties and instances. Figure 8.16 presents object properties used to structure the record for *Patient1* (instance of *Inhabitant* class). For instance, *Patient1* individual can have an object property of "*hasMedicationsList*" and value as an object instance of "*Patient1_Medicines*" (bucket). This bucket, ("*Patient1_Medicines*") can have *N* number of medicines object instances as a value, such as "*Medicines_MemantineDrug*", which is defined using the sub-property of "*hasMedicineList*" object property called "*hasMedicineListItem*". The individual, "*Medicines_MemantineDrug*", will hold all the relevant data required for the medicine, such as the description of the medication and instructions of dose timing. This process can be repeated to represent other application scenarios, such as doctor's appointments lists, notifications services, and other user-specific preferences.

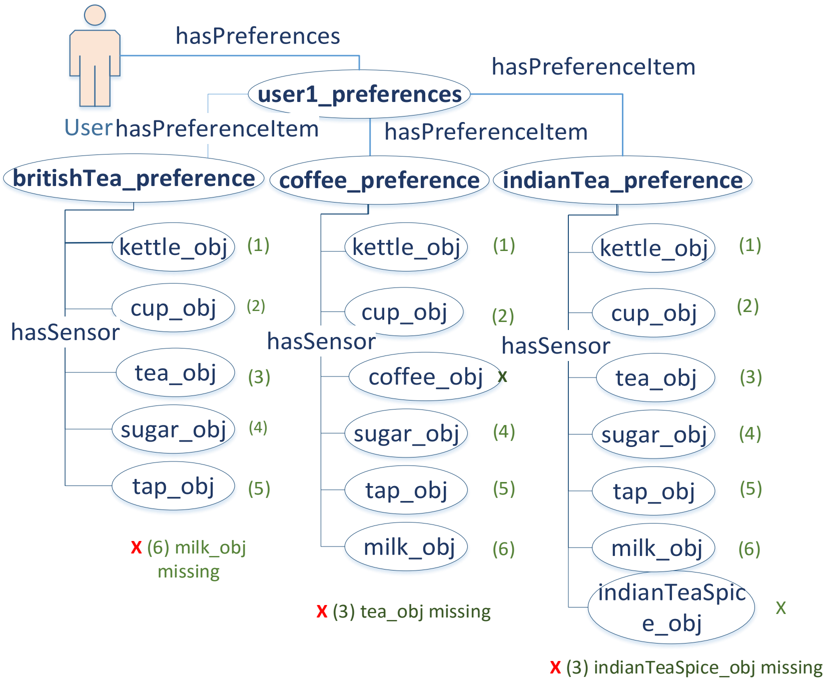
8.4.4.2. Single-user SPARQL-based Inferencing

In order to perform activity assistance in ADL, a simple simulated environment is created to enable various sensors and view the AR results (see Figure 8.19); here, the Text-to-Speech feature is also used for the resulting output. The AR inferencing is performed by the web service using SPARQL queries. For this, only pre-defined user preferences [shown in Figure 8.7(a) and Figure 8.8 for the preference management interface] are applied to match against the activated sensors. The aim of the matching process is to find the related user preference(s) and other inactivated sensor object(s) from the matched individual preference(s) in order to complete the activity. For this, the following steps are followed to perform SPARQL queries.

1. Find a user preference that has all the activated sensor objects and does not contain additional sensors objects in the same preference.
2. Otherwise, N number of user preferences are returned, which has all, or some activated devices listed in a particular preference and other inactive sensor objects.

2.1.1. The number of activated sensor object(s) exist in each user preference is taken and ordered in descending order.

2.1.2. Using the results obtained, the search for the missing sensor object(s) is carried out by inspecting the individual user preferences. The matched sensor object from the individual user preference is excluded by using the key functions, such as FILTER, Logical & Comparisons, or Conditional SPARQL operators [139].



Example

Activated Sensors : (1) kettle_obj, (2) cup_obj, (3) tea_obj, (4) sugar_obj, (5) tap_obj and (6) milk_obj.

Step 1 –> no exact matching preference.
Step 2 –> all three user preferences names are returned.

Result : **User Preferences**	**Step 2.1** (match count)	**Step 2.2** (missing from preference or activated sensors list)
britishTea_preference	5/6	X (6) milk_obj
coffee_preference	5/6	X coffee_obj - not in activated sensors list
indianTea_preference	5/6	X indianTeaSpice_obj - not in activated sensors list

Figure 8.18. Illustrating the inferencing steps taken using the SPARQL query language

Figure 8.18 illustrates the above steps to perform SPARQL based in activity inferencing based on the preferences defined in the model at the top of the figure and worked example at the bottom. The worked example show that six sensors were activated out of three user preference defined for making tea in the model; each had one missing sensor as listed in the table for step 2.2. The key benefits of this SPARQL query-based approach are that no model loading or reasoning libraries are required. Nevertheless, this approach does require explicit relationships to be defined in the dataset. To bridge this gap, the notion of SPARQL Inferencing Notation (SPIN) can be used to create rules, constraints, and functions in SPARQL syntax, which can be executed on the triplestore. SPIN is also known as SPARQL rules; for more information, see [161], [264].

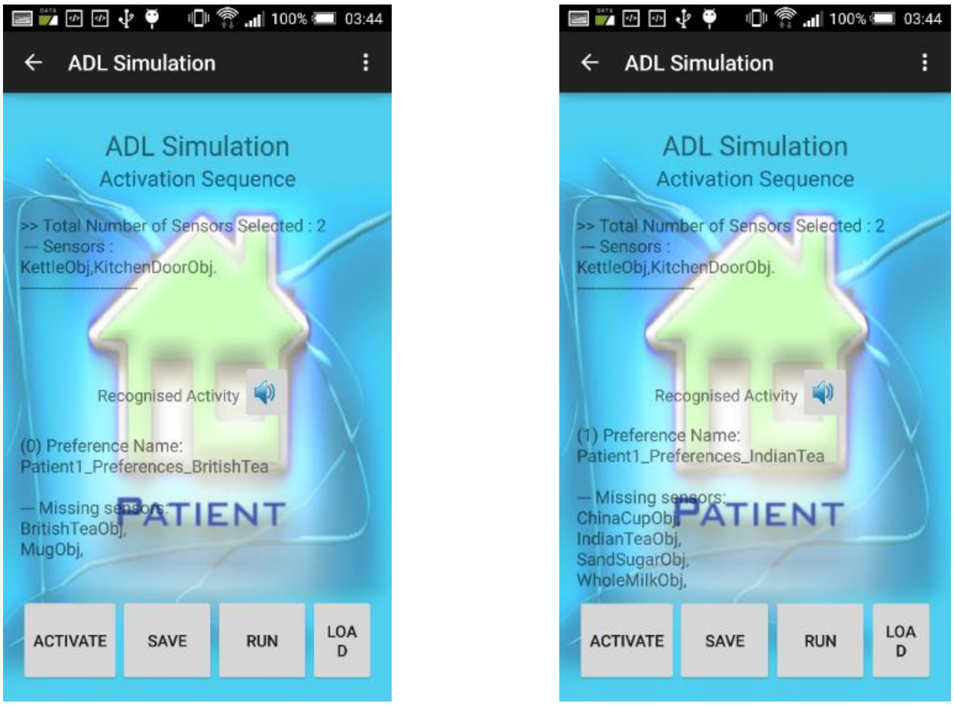


Figure 8.19. ADL simulation result of two possible preferences with their missing sensors to complete the activity

8.5. Evaluation and Discussion

A smart lab infrastructure was developed with real-time sensing environment in two stages. At each stage, three main experiments were performed. The first experiment was conducted during the first stage where only binary sensors were used to conduct SPARQL based inferencing; details in section 8.5.1. The second experiment was also conducted in stage one to evaluate semantical segmentation algorithm and the results were presented in CHAPTER 3. The third experiment was conducted at the second stage of smart lab infrastructure development with multimodal sensors embedded within everyday objects to detect fine-grained actions of the users. The preliminary result of the third experiment results was presented in CHAPTER 4. Further development effort is required to complete the integration of single-user AR framework and multi-user AR proposed in CHAPTER 6 and CHAPTER 7. However, to illustrate the applicability of the proposed MSA and real-time multimodal smart kitchen environment, a typical multi-user scenario and AR steps is presented as a case study in section 8.5.2. Furthermore, open issues and challenges faced during the development of MSA and real-time smart environment are discussed in section 8.5.3.

8.5.1. SOA: Querying-based Inferencing Experiment

In experiment 1, the SPARQL based activity recognition algorithm presented in section 8.4.4.2 is evaluated with binary sensing environment depicted in Figure 8.12. The time duration between sensor activation and generation of inferencing results on the client device is measured to assess the overall system performance. The sensor activation time is only registered once the data are received by the web service to reduce factors such as network delays, time synchronisation between the sensing devices.

Table 8.2. User activity preferences with the associated total number of sensor objects

Activity Number	UAP	Sensor Objects Sequence	Total no. of sensors
1	Make Indian Tea	KitchenDoor1, KitchenCupboard1, TeaBagJar, IndianTeaSpiceJar, SugarJar, Kettle1, KitchenTap1, Fridge1, MilkBottle1, EatingSpoon1, Mug1	11
2	Make Cappuccino Coffee	KitchenDoor1, KitchenCupboard1, CappuccinoBagJar, SugarJar, Kettle1, KitchenTap1, Fridge1, MilkBottle1, EatingSpoon1, Mug1	10
3	Make Strawberry Juice	KitchenDoor1, KitchenCupboard1, JuicerMixerCup1, SugarJar, KitchenCupboard2, ChoppingBoard1, Knife1, Fridge1, StawberryPacket1, MilkBottle1, KitchenWaterTap1, GlassCup1, JuicerMixer1	13
4	Making Chips And Beans	KitchenDoor1, FridgeFreezer1, ChipsBag1, KitchenCupboard2, OvenTray1, HeinzBakedBeansCan1, KitchenWaterTap1, MicrowaveBowl1, OvenDoor1, MicrowaveDoor1, CeramicPlat1	11
5	Make Pasta	KitchenDoor1, KitchenCupboard1, PastaBag1, PastaPot1, KitchenWaterTap1, WoodCookingSpoon, PastaSauce, SaltBottle1	8
6	Taking Medicine	KitchenCupboard1, MedicineContainer1, GlassContainer1, KitchenWaterTap1	4

Note: [sensor] - Changes in object(s) from previous activity

Table 8.3. AR test scenario types

Scenario Types	Exact no. of Sensors	Extra Sensors Activation	Faulty/ Missing
TP1	✓	×	×
TP2	×	✓	×
TP3	×	×	✓

Table 8.4. Two examples of AR test cases

#	Examples of tests specifications
1	TP1: #1, TP2: #1, add *KitchenCupboard2* and *GlassCup1*. TP3: #1, swap *TeaBagJar* and *OvenDoor1*.
2	TP1: #2, TP2: #2, add *KitchenCupboard2* and *GlassCup1*. TP3: #2, replace *Mug1* with *GlassCup1*.

A fixed time window length is defined for six user activity preferences (UAPs) that are listed and tested with three different scenarios, see Table 8.2 and Table 8.3. The first scenario (TP1) activates the exact number of sensors defined in the user preferences, the second scenario (TP2) shows the activation of additional sensors objects, and the third scenario (TP2) shows a simulation of faulty sensors by using some sensor objects that are missing or not activated. The scenarios for the first two activities are illustrated in Table 8.4. Overall, each of the six activities is executed with three different scenarios by two actors (Exp).

The web service was deployed on the HP Z440 workstation with Intel(R) Xeon(E) v3 3.50GHz processor with 16GB RAM. The mobile application was tested on a Samsung S6 edge smartphone running Android 6.0.1 OS. The sensing data were collected using several touch sensors and door contact sensors using varied protocols.

The results in Table 8.5 indicate that on average, it takes 4477ms to receive the inferencing result on the mobile phone for all six UAPs with three different scenarios executed thrice. Overall, the results show little to no correlation between the number of sensors in the UAPs and the average time taken for inferencing and then communicating the results to the user.

Table 8.5. Results showing average activity inferencing duration from the last activities recorded

#	Test Type	Exp1 (ms)	Exp2 (ms)	Exp2 (ms)	Avg. (ms)	Avg. Per # (ms)
1	**TP1**	3890	3988	5127	4335	
	TP2	5175	4176	4802	4718	**4472**
	TP3	4172	4145	4776	4364	
2	**TP1**	4013	3953	4439	4135	
	TP2	4131	4135	4725	4330	**4288**
	TP3	4275	4288	4630	4398	
3	**TP1**	3926	3923	4353	4067	
	TP2	4303	4316	4571	4397	**4411**
	TP3	5310	4225	4768	4768	
4	**TP1**	4116	4175	4452	4248	
	TP2	6330	4474	4695	5166	**4636**
	TP3	4410	4461	4614	4495	
5	**TP1**	4150	4265	4409	4275	
	TP2	4446	4414	5919	4926	**4584**
	TP3	4497	4533	4624	4551	
6	**TP1**	4166	4801	4271	4413	
	TP2	4532	4556	4563	4550	**4474**
	TP3	4415	4460	4498	4458	
						4477

8.5.2. MSA: Multi-user AR Case Study

In general, an activity is recognised by inspecting each action at two granularity levels, coarse and fine-grained. Figure 8.20 depicts a typical kitchen (S1) and living room (S2) ADLs actions conducted by Bob and Alice in a multi-room and single-room. At coarse-grained action level, general context, relations between ADL descriptions and user's actions are used to assume an activity unfolding, i.e., *cup* (S7), *kettle* (S3), *tea bag* (S6), *milk* (S5), *sugar* and *tap water* observations for *MakeTea* ADL. Whereas, fine-grained level action detection method inspects deeper on how each action for a specific ADL is performed and determines whether the intention of a given action is satisfied. For instance, detecting "*filling up*" kettle from the water tap, "*pouring*" water from the *kettle* into a cup and "*drinking*" from the *cup* when conducting *MakeTea* activity.

In order to understand who is conducting actions for a given ADL in a shared environment, multi-user AR approach need to initially detect if the actions are occurring in multiple locations at the same time interval or assuming that a user cannot interact with more than two objects at a given time interval (T1-T3). Therefore, enabling parallel activities occurring in multiple rooms to be detected in T1 where Bob (S1) is filling up kettle (S3) in the kitchen and Alice(S2) is using TV remote (S4) in the living room (S2). Similarly, parallel activities in T3, drinking and turning the microwave on actions for different activities can be detected in a single room. The collaborative activity in a single room during T2 can be detected using discriminative sensors and approximating action association to a given user.

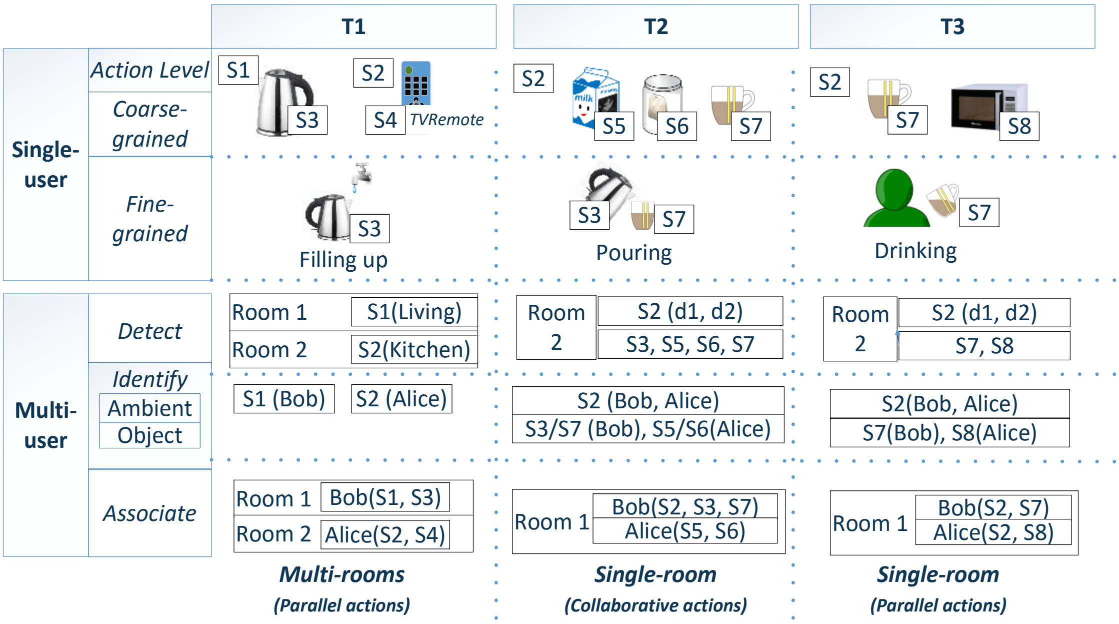


Figure 8.20. An example of multi-user activities in a shared kitchen and living environment

Apart from the technologies already mentioned in the previous sections, other supporting software components that are used to build the system are Jersey libraries [193] (i.e., Jackson library for JSON strings to java object mapping), Jena [265] Pellet [190](reasoner, see others [155]), Protege [159] (ontology editing tool), and Google API Services [266]) (i.e., for Text-To-Speech APIs, and Maps API. The Jersey library plays a key role in developing the RESTful web services for the function and parameter mappings of the incoming requests from the clients, as well as in producing and consuming data in various formats dynamically. In general, Jersey library is used to bind the web services with the Android application and mapping data into various object classes.

8.5.3. Discussion

The past system implementations with similar architectural styles and patterns to MSA have shown positive results in both functional and non-functional requirements; not only for AAL systems [267], [268]. However, finding suitable design patterns for a given application can be challenging and be easily misused [132], [134]. Nevertheless, several benefits of using a popular style and pattern exist. One example is system maintainability, which can improve code compensation level and efficient debugging for the developer. Furthermore, the decomposed MSA can enable any application to improve its scalability. In the case of the proposed system, additional multimodal sensing devices can be added within the SH so that the server can efficiently collect, process and disseminate data to multiple clients with minimal delay. Moreover, creating an opportunity to integrate other third-party services to extend the sensing capabilities of the smart environment.

Several lessons were learnt during the development of MSA and complex smart environment with open-source, off-the-shelf and bespoke sensing devices. Five key lessons learnt are as follows. Firstly, decoupling AR tasks with multiple web services can create additional overhead and duplicate codes. In addition, data analytics tools and hardware communications libraries require a large number of dependency libraries which are challenging to manage, outdated, not maintained, and not compatible with other libraries. Fortunately, microservices framework Spring book and dependency management tools such as Apache Ant, Apache Maven and Gradle are available to efficiently manage the dependency libraries and set up a new development environment more efficiently. Using these tools can prove to be more advantageous when integrating more third-party libraries and APIs to collect data from new smart home devices.

Secondly, developing bespoke sensing platform using IoT platforms or microcontrollers or using off-the-shelf sensing platform, it is essential to synchronise all the platforms to the same time server. This task can be tedious and challenging to achieve if the source code for the third-party library is not available. Nevertheless, data collected from the unsynchronised sensing platform can provide impact the accuracy of AR algorithms as the timestamp on the data can fall under different time interval or window-size. More specifically, impacting the ability to fuse multimodal sensor data from platform 1 compared to platform 2, which difference of even one minute in the internal clock. Moreover, re-programming/updating bespoke sensing devices with several modules deployed in the environment can be challenging to dismantle or unplug from objects physically. Therefore, over-the-air programming features available on some of the IoT based microcontroller should be utilised to avoid physical tasks.

Thirdly, low energy devices that are wireless generally have limited battery lifespan. Therefore, self-recharging capabilities using natural environments such as light, wind, heat and kinetic energy options should be explored to avoid replacing batteries every couple of hours or days. Moreover, a large amount of energy can be conserved when transmitting the data over a network to the cloud servers to perform some basic data analytics that edge devices are now powerful enough to compute themselves. Therefore, future work should explore edge and fog computing paradigms to not only reduce the energy but also increase the availability of the system.

Fourthly, a higher number of processing cores and optimised graphic-cards are required to support each of the AR tasks of data collections, storing, reasoning and disseminating results. Some AR tasks may require processing mathematical calculations and other more threads to complete the tasks. Therefore, intelligent thread prioritisation, scheduling and parallel processing are necessary to optimise and develop real-time AAL system.

Finally, the HCI with the system plays a key role in gaining further benefits from the system's capabilities. The system implementation uses a mobile application; however, our society is moving towards more natural and ubiquitous HCI. Other systems discussed [248], [250] in section 8.2, have already adapted the notion of augmented reality to overlay instructions on the camera or use natural gesture-/voice-based HCI. In comparison to the standalone/SOA SMART system and other systems implementations discussed in section 8.2, mainly have a web-browser based interface, this may limit the client devices from further utilisation, unlike with mobile devices with embedded sensor capabilities to collect meaningful and contextual data. In addition, instead of configuring additional dense or ambient sensors in the SH environment, more external sensors can be directly attached to a mobile device using any standard communication protocol [269].

8.6. Summary and Future Work

This chapter presents a literature review on system architecture styles and patterns adapted by recent AAL facilities and the challenges faced developing it. This was achieved by reviewing some of the latest studies and AAL system components that can complement one another. Based on the findings, a microservices-based architecture (MSA) approach is proposed for an AAL system. MSA integrates and extends the capabilities of the previous system implementations by introducing a light-weight, REST-based web service with an Android mobile application and web-browser interface. Multiple interconnected microservices plays a key role in interacting with the triplestore endpoint, collecting data from SH sensors and providing information to client applications. The service API web service provides activity inferencing and reasoning capabilities using Jena API; different reasoning engines can also be easily integrated. In addition, the service API web service has the ability to integrate rules-based reasoning methods such as JESS, SWRL/SPIN as they are based on Java and supported by Jena API. To model ADL knowledge and recognise user activities, Semantic Sensor Network (SSN) vocabulary was reused to describe uncertainties and imprecise knowledge in representing heterogeneous sensing platforms, smart environment, multi granular ADL actions and performing AR.

The proposed MSA approach was developed in a smart lab environment with real-time sensing environment and conducted controlled experiments in well-designed activity scenarios to create a dataset. A dataset with three kitchen-based activities, one taking medicine and one living room-based ADL was collected over a week. The dataset contains single-user action conducting actions with individual objects that have multiple embedded sensors attached to them to allow detection of human interactions at multi granularity level. In addition, the open data initiative (ODI) guidelines have been followed to make the dataset publicly available with tools to enable the research communities and industry partners to compare their algorithms more efficiently.

A real-time MSA and multi-layered service-oriented architecture (SOA) system prototypes were implemented to assess the feasibility of the proposed software and hardware architecture. The results of an SOA with a binary sensor environment show that the average SPARQL inferencing time taken to display the results to the user is 4477ms on average. Moreover, SOA was also evaluated for semantic-based segmentation in 2.8 and showed the performance has suffered to segment each event with the average classification time of 3971ms and 62183ms for single and mixed activities scenarios, respectively. SOA further suffered in the realm of performance, maintainability and availability of the system when analysing computationally demanding tasks such as fine-grained level AR recognition and evaluating uncertainties in HAR. On the other hand, the initial development effort of the MSA approach demonstrated that continuous real-time heterogeneous multimodal sensors and communications protocols could be used to collect and perform AR tasks in a reasonable time.

Finally, MSA shows greater flexibility and potential to be further developed in terms of usability, ability to support additional application scenarios, and capacity to provide a greater scope of collecting personalised and contextual data, thus increasing the accuracy of AR. More specifically, the future implementations will focus on areas such as activity learning, improving data modelling techniques, semantically processing raw sensor data with an efficient timing mechanism[18], evaluating multi-user AR approach presented in CHAPTER 7, as well as enhancing the SH sensing capabilities, performance optimisation, and HCI methods (i.e. utilising Amazon's Alexa voice services [122]). Furthermore, exploring rules (i.e., SPIN [270] and SWRL rules [271]), and Description Logics (DLs) capabilities instead of current SPARQL-based querying approach can be carried out to improve performance. Subsequently, to test the AR and learning algorithm, create and benchmark multi-user dataset with multimodal sensor data to compare against state-of-the-art approaches. However, more efforts are required to create analytical tools for the developers, such as automatically formatting datasets and semi-automatic annotation approaches [272].

CHAPTER 9. CONCLUSIONS AND FUTURE WORK

The rising global ageing population is a positive advancement for the human race. However, we are now faced with new challenges in our societies to manage social and economic problems related to the elderly population. The elderly population have a higher probability of experiencing physical and cognitive decline, which hinder their ability to performing activities of daily living (ADLs) independently. Hence, creating a significant impact on the overall quality-of-life of the elderly. Therefore, the opportunities created with a technologically based solution is widely investigated by academic, healthcare providers, commercial industries and other stakeholders globally to provide just-in-time and context-aware assistance to individuals in their desired environment, i.e., at home. For this, ambient assisted living (AAL) system are developed with unobtrusive sensing environment (i.e., smart home (SH)) to monitor an individual's engagement and provide activity-aware services. Nevertheless, to provide reliable, accurate and effective activity-aware services, activity recognition (AR) plays a critical role in the AAL system.

This chapter summaries research conducted in section 9.1, highlight key findings and contributions of the thesis in section 9.2, shed light on further research directions in section 9.3 and provide concluding remarks in section 9.4.

9.1. Summary of Work

CHAPTER 1 introduces the research background, motivation for investigating in assistive technology for the ageing population and highlights the problems that this study intends to solve. It also outlines the overall aims and objectives of this thesis and presents the research methodology and scoping techniques for achieving the aims.

In order to achieve the aims and objectives, a review of existing literature is conducted to analyse the strength and weaknesses of previous and state-of-the-art studies, identify challenges and open issues in AR tasks and overall in AAL system is presented in CHAPTER 2. The key challenges identified in the literature include accurate AR, mixed activities recognition, real-time continuous sensor data segmentation, ambiguity in non-binary sensor data, uncertainty factors resulting in missing data or becoming unreliable and selecting appropriate system architecture style to perform several resource-intensive AR tasks. The subsequent chapters present approaches and technical work to address these challenges.

Objective 1. was to develop an activity model which can be reused for inferring and recognising mixed activities conducted by single or multi-users in a shared environment. Therefore, knowledge-driven (KD) approach and semantic technologies are leveraged to model different aspects of ADLs being conducted in a given smart environment. This activity model is the backbone of the chapters 3-7 and each chapter enrich this activity model to express ADL and AAL specific knowledge at multiple levels of abstraction in order to address specific challenges identified from the literature review such modelling fuzzy concepts and uncertainties factors.

Objective 2. was to conceive and develop a semantic-enabled algorithm to disentangle the sensor observation from real-time continuous sensor data stream into a relevant set of ongoing ADLs. CHAPTER 3 address this challenge and fulfils the objective 2. by adapting a knowledge-driven (KD) approach and proposing a semiotic theory inspired ontology-based knowledge modelling and reasoning approach. In addition, this approach incorporates personalised user actions when conducting a given activity. This approach was evaluated using 30 simulated use scenarios which include sequential and mixed activities. The result showed that sensor observations were segmented with minor improvement in accuracy for single and mixed activities scenarios. One of the main limitations of the ontology-based approach is that only binary relations and binary sensor data (also referred to as crisp knowledge) can be represented and reasoned.

Subsequently, for objective 3. , the segmented set of ADLs containing both binary and non-binary sensor data are used to recognise single-user AR at coarse- and fine-grained action level in CHAPTER 4. For this, crisp and fuzzy ontology-based knowledge modelling and reasoning approach were proposed with multimodal sensing attributes required to recognise completion of user actions at a fine-grained level accurately. The main challenge addressed in this chapter is to model fuzzy concepts using fuzzification method, fuse multiple sensor attributes using fuzzy rules and performing defuzzification based on the sensor's input at a given time instance to detect fine-grained actions based on conditions defined in the fuzzy rule. The approach was implemented using fuzzy ontology plugin in Protégé and fuzzyDL reasoner. In addition, threshold values required to create a fuzzification model from imprecise sensor data, datasets collected from the heterogeneous multimodal environment created in the smart lab. These threshold values from the dataset were further used to evaluate the fuzzy ontology-based fine-grained AR algorithm. In order to evaluate, an experiment based on three fine-grained actions required to make tea activity using multimodal sensor data from the dataset. The preliminary result shows the usefulness of approaches to detect user actions with an object using multimodal sensing attributes at the fine-grained action level. However, further work is required to address the scalability, maintainability and performance optimisation that come with limited or underdeveloped tools available to model fuzzy knowledge.

Objective 4 focuses on uncertainty knowledge modelling and reasoning in CHAPTER 5. CHAPTER 5 initially review uncertainty theories, probabilistic, evidential and fuzzy, which can be adopted in the ADL knowledge modelling and reasoning algorithm. As a result, probabilistic based uncertainty modelling and reasoning was proposed with the consideration of four key uncertainty factors: technology, human, object functionality, and environmental. Therefore, probabilistic ontology (PR-OWL) based on uncertainty knowledge modelling and reasoning is proposed. PR-OWL is founded on first-order logic (FOL) and Multi-Entity Bayesian Network (MEBN) to model implicit joint probability distribution over a likely unbounded number of uncertainties. PR-OWL enable situation-specific Bayesian Network (SSBN) based on pieces of evidence collected from the smart environment and propagate the joint probability tables for the effected uncertainty factors. Moreover, an uncertainty reasoning algorithm is proposed with a case study to recognise model and reason with four types of uncertainty factors at activity and one fine-grained action specific level uncertainty reasoning. The findings from the case study suggested the applicability of PR-OWL based uncertainty reasoning and lay a foundation for future work for integrating data-driven approaches to create a hybrid approach to learn and evolve the initial knowledge model over time.

Objective 5 was to create an approach to incorporate imprecise knowledge and factors of uncertainties within the single-user AR. Subsequently, CHAPTER 6 proposes a framework to incorporate crisp, fuzzy and PR-OWL for knowledge modelling and reasoning in a unified single user AR process. The process involves multiphase knowledge development, mapping and reasoning between separate ontologies. The separate ontologies are created due to the incompatibility of the tools, complexity of information types and abstraction level required to model ADLs at the coarse and fine-grained action level. The main benefit of this framework is that it enables separate ontology models are created to allow knowledge to be lightweight, easy reused, maintain and trace over time. Moreover, this chapter presents a method which interprets fuzzy sensor data to detect incomplete user actions and missing user actions using Allen temporal rules. Next, an algorithm for single-user AR process is developed using the ontological modelling framework. To evaluate the framework, all three types of ontology models are developed and mapped for a *making tea* ADL, *pouring* fine-grained action with the kettle and relevant four types of uncertainties. The initial finding suggests the applicability of the framework to incorporate crisp, fuzzy and PR-OWL knowledge in single-user AR. The future work will involve applying this single-user AR framework into the real-world environment, testing the approach under various conditions and comparing the results with other studies.

Objective 6 was to conceptualise and develop an approach for multi-users AR ($\mathcal{MAR}$) within a shared living environment. CHAPTER 7 reviews recent studies to identify challenges in recognising multi-user activities in a shared environment. The key challenges identified was to detect, identify and associate user's actions. Hence, an ontology-based framework for single user AR $\mathcal{MAR}$ approach with time-series analysis/location information and discriminative sensors approach is developed to detect, identify and associate individual actions in a shared environment. In order to associate multiple user actions in shared space, fingerprint sensor observations can identify and allow direct association with an object and radio frequency identify (RFID) tags for indirectly associating the user action based RSSI and proximity information of an object. Moreover, a method to estimate AR confidence level ($\mathcal{ARCL}$) for ongoing activities at coarse- and fine-grained action level is developed. One key benefit of this approach is that only ambient sensors and embedded sensors for a non-invasive and non-obstructive data collection approach are proposed. $\mathcal{MAR}$ is applied to a kitchen and living room application scenario to illustrate its use of the approach. However, future work will use the multimodal sensing environment to create a multi-users dataset to create a benchmark and compare the multi-user AR algorithms performance with other studies.

The objective 7. was to investigate and develop an appropriate system architecture for AAL system and SH technologies, which is, interoperable, reusable, flexible, expandable, scalable and more maintainable. CHAPTER 8 reviews recent AAL systems from the system architectural perspective and identifies service-oriented architecture (SOA) is now commonly adopted. Additionally, optimised graph-based database over the traditional relational database is being integrated due to the ability such as expandable linked data, improve readability and query time optimisation. Therefore, microservices-based system architecture (MSA) is proposed to share AR tasks with five key web services configured on separate machines that can collaboratively perform respective tasks. The first web service communicates between client devices and four internal web services for data collection from a smart environment, big data storage, data processing services and application-specific assistive features. The proposed MSA is a subset of SOA, and it has emerged from progressive development, evaluations from a single web service based on demand for high accuracy and performance for AR tasks from the aforementioned objectives. This chapter provides implementation details of the system used for evaluating the proposed approaches and algorithms in previous chapters using under various kitchen-based activities and scenarios.

Finally, objective 8. was to disseminate the findings after identifying AR-related challenges, proposing and evaluating the novel approaches, frameworks, methods and algorithms to the broader community. Four journal papers, four conferences, one book chapter were published, one journal paper currently under review and several seminars were provided at international avenues to disseminate the findings from this thesis. The full list of publications can be viewed in detail on page v.

9.2. Summary of Contributions

The research presented in this thesis makes advances addressing single and multi-user AR challenges from activity modelling to data collection and reasoning with the imprecise and uncertainty knowledge. As a result, seven key contributions have been made in this thesis. These seven contributions are: (1) multi-layered ADL knowledge model containing crisp, fuzzy and probabilistic knowledge to recognise activities at multi-granularity action levels, (2) semantic-enabled data segmentation, (3) fuzzy-based fine-grain activity recognition, (4) uncertainty with probabilistic reasoning, (5) single-user framework to handle imprecise and uncertainty knowledge, (6) multi-user AR approach, and (7) microservices-based system architecture (MSA) within real-time smart environments.

The first contribution is a *unified multi-layered ADL knowledge model containing crisp, fuzzy and probabilistic knowledge* to support recognition of single and multi-user activities at fine-grained action level within the context of AAL and SH domain. Knowledge-driven (KD) mixed activity modelling has received little attention in the past due to high computation requirements and limited ability to express imprecise and uncertainties with an ontological modelling approach. However, with the advancement in high-performance computers and recent efforts by Umberto Straccia and Rommel Carvalho to extend OWL's expressivity with fuzzy set theory and probabilistic uncertainty theory in ontological modelling, respectively, has paved the way for a number of real-work applications. Hence, this ADL model for AAL system adapts their fuzzy ontology and probabilistic ontology (PR-OWL) modelling tool to encode crisp, imprecise and uncertainty knowledge in three main layers. The first layer develops crisp knowledge containing descriptions of the ADLs, environmental entities, diverse sensor network, user profile and other application-specific information. Subsequently, external Semantic Sensor Network (SSN) vocabulary is imported in this model to comprehensively describe a complex sensor network with attributes such as operating conditions, sampling and data storing procedures. The second layer of the ADL knowledge consists of fuzzy ontology used to describe imprecise sensor data and conditions under which fine-grained actions for each ADL is complete. Lastly, uncertainties caused by human, environmental, technological and object functionality factors are encoded in the PR-OWL with joint probabilities indirectly affecting the AR results at activity and action level.

The second contribution in AR is the *semantic-enabled data segmentation algorithm with user preferences* of observed sensor events when ADLs are performed in a simple or mixed activities scenario. Several studies have proposed methods of separating and organising sensor observations by first storing and querying from a database and then reasoning with a generic description of ADLs. However, little has been explored in semantically distinguishing individual sensor events with the knowledge of user preferences to directly segment to the relevant ongoing/new ADLs. Hence, the semiotic theory inspired the ontological model, capturing generic knowledge and inhabitant-specific preferences for conducting ADLs to support the segmentation process is proposed.

The third contribution is the *fuzzy ontology-based approach is to recognise actions at a fine-grained level* is to support Parkinson patients suffering from tremor to detect mishaps/spillage/dropping of an everyday object and Alzheimer patients forgetting to use the object after the initial interaction. In this thesis, binary interaction with an object is considered as coarse-grained level action recognition, whereas, an object attached with multiple non-binary sensors to analyse the usage of an object as fine-grained level action recognition. The non-binary sensor measurements are inherently imprecise and subjected to individual interpretations or the material nature/dimensions of given objects. Hence, this study explores the fuzzy ontology modelling method to define imprecise sensors data in gradient values, fuzzy rules to define fine-grained actions and defuzzification method to reason with the raw data input. The fuzzy ontology is based on Fuzzy set theory which was initially introduced by L. Zadeh in 1965 and applied in ample of real-world systems and devices. Nevertheless, limited studies have explored the applicability fuzzy ontology reasoning in detecting action at the fine-grained level in a real-time smart environment.

The fourth contribution is the *probabilistic ontology-based modelling and reasoning method* to define uncertainty factors in activity recognition. Four key factors that influence the result of the AR are a technological failure (i.e., sensor failure, low battery), object malfunction, and human errors (mishaps, malicious actions and forgetting actions). Based on the findings of current literature, probabilistic (Probabilistic theory), belief (Weighted Average Combination, Type-2 Fuzzy Logic), rules (Allen Temporal Logic, Dempster-Shafer) and network (Bayesian Network, Markov Logic Network) are amongst the conventional approaches to handling uncertainties. This study further extends, the probabilistic theory by leveraging Probabilistic ontology (PR-OWL) modelling and reasoning method to evaluate the uncertainties when recognising ADLs. Multi-entity Bayesian Network (MEBN) theory is the core component of the probabilistic knowledge modelling process to capture the four uncertainties factors. Moreover, a framework is proposed to combine, fuzzy ontology and PR-OWL to interpret/fuse imprecise non-binary sensor data to recognise fine-grained actions and anticipating un-/known uncertainty factors.

The fifth contribution is a *single-user AR framework* which incorporates *crisp, imprecise and probabilistic ontology knowledge modelling and reasoning process*. This framework enables activities to be recognised at the coarse- and fine-grained action level with their respective uncertainty factors. AR at coarse-grained action level can recognise candidate activities occurring with basic relations with objects and ADL in the crisp ontology and Pellet reasoner. In addition, missing actions are detected at a coarse-grained action level by defining a set of mandatory and dependency actions and analysing time-series data using Allen temporal rules to detect missing actions. On the other hand, AR at fine-grained action level, incomplete actions with an object is detected using fuzzy ontology and fuzzyDL reasoner. Next, the uncertainties factors impacting AR results are applied at both levels are defined and propagated as the evidence from the smart environment unfolds to create SSBN. Based on this single-user AR framework, an algorithm is developed, and evaluation results are discussed.

The sixth contribution is the *multi-user AR approach* that identifies and associate user actions in a shared smart environment. The approach leverages a combination of location, time-series and discriminative sensing (fingerprint & RFID tag) to identify the number of inhabitants and associate their actions with everyday objects in a shared smart environment. The application of this approach is to support personalisation applications for inhabitants when conducting ADLs, monitoring and learning their change of behaviour over time.

Finally, *MSA tailored for AAL, and multimodal SH hardware architecture* with open-source, off-the-shelf/bespoke sensing techniques is proposed. The existing standalone, enterprise service bus (ESB) based SOA and other architecture styles adapted for AAL systems in the literature is first analysed. Based on the findings and requirements of the AR tasks, a novel MSA is proposed with the integration of the latest semantical technologies and tools. At the software architectural level, multi-layered REST-based web services with application programming interfaces (APIs) are developed to perform dedicated tasks such as data collection, storage, processing data and application level. Moreover, tools and libraries required to support three types of complementary ontologies (crisp-OWL, fuzzy-OWL and PR-OWL) reasoning algorithm were integrated into the system. At a hardware architectural level, a smart lab environment was developed with multimodal off-the-self and bespoke ambient and embedded sensing approach. In general, this system prototype was used to test and evaluate the approaches presented in the thesis with kitchen-based test case scenarios. Moreover, an Android mobile application and web-browser based user interfaces were developed to enable client devices to communicate with the web service API for the sensor data, obtain AR results and even configure sensing environment. Furthermore, the prototype provides other supportive utility tools such as *simulator/synthetic ADL data generator* for the experimentation and converting dataset from JSON/XML to *support open data initiative (ODI) framework*[108] for efficient data sharing.

9.3. Open Issues and Future Work

There are several opportunities, issues and open problems identified during this research. The proposed solutions for a given problem need to be investigated further and applied in the real-world application as it may suggest a new and interesting set of problems. Throughout this thesis, several future directions of this research were highlighted. However, some of these are highlighted below for future research.

Firstly, semantical segmentation achieved high accuracy with ontology-based terminology box (T-box) and assertion box (A-Box) reasoning using incremental Pellet reasoner. However, using description logic (DL) querying method on the ontological model may result in optimising the segmentation performance time. Hence, comparing both, Pellet and DL querying based approach for accuracy and performance trade-offs can give more sights to create an efficient segmentation algorithm. In addition, high-performance time-series graph-based databases such as pipelineDB [251] need to be investigated in order to enhance the semantical segmentation accuracy and performance further.

Secondly, the proposed segmentation approach currently provides a mechanism to allow users to specify their preferences when conducting an ADL by selecting the instance of sensors attached to everyday objects. However, more control and enrichment in single and multi-user AR approach is required for personalised assistance based on user profiles (i.e., reminding diabetic patients to add sweetener to their drink/food instead of sugar).

Thirdly, the use of fuzzy ontology-based AR approach for fine-grained action level was evaluated on a small set of actions and fuzzy rules. However, more investigation is required to scale the fuzzy rules for a bigger sample set of fine-grained actions to be detected with a real-time sensor data stream. Similarly, the future research direction for uncertainties reasoning with PR-OWL is to be embedded into the MSA based AAL system prototype with additional factors affecting the recognition of activities and actions within real-world smart environment context.

Fourthly, the use of fuzzy ontology and probabilistic ontology knowledge model in single-user AR framework requires further investigation in terms of simplifying modelling process with the aid of a single tool for crisp, fuzzy and uncertainty modelling. In addition, reasoning performance with each type of knowledge was noted to take a considerable amount of time on a standard machine. Hence, fine-tuning is required on the three knowledge models, mapping and algorithm, in order to run three types of reasoning on continues sensor data stream.

Sixthly, creating a benchmark for datasets containing single and multiuser activities conducted in a real smart environment with time synchronised multimodal sensor data. This dataset will be used to evaluate the single and multi-user AR approach developed using knowledge-driven (KD) approach and comparing with existing data-driven (DD) studies in the literature. Based on the comparison result, the capabilities of the KD and DD approaches will need to be evaluated and develop a hybrid activity learning approach. The purpose of a hybrid activity learning approach will be to identify frequency/patterns and evolving the initial domain-specific knowledge over time.

Lastly, proposed MSA rely on cloud computers to perform all AR-related tasks such as storing and reasoning with all the smart environment data transmitted over the network. This MSA approach requires a large amount of energy to transmit the data, creates delays on the network infrastructure and processing unnecessary data which can be filtered by the edge devices before transmitting. Hence, making inefficient use of cloud computing resources. Therefore, investigation in edge and fog computing paradigms within AAL systems is required to better utilise the hardware capabilities located near the sensing device and manage cloud computing resources more effectively.

9.4. Concluding Remarks

The research presented in the thesis makes a significant contribution to knowledge based on overall aims and objects to advance in challenges faced in HAR for AAL systems. More specifically, this thesis makes critical advances in developing KD methods, approaches and framework to address challenges such as semantical segment sensor data with user preference management, recognise single/multi-user activities at fine-grained actions level using fuzzy set theory and handling uncertainties with probabilistic theory. In-depth literature reviews in the aforementioned areas were carried out and highlighted the challenges and open issues. Based on the findings, six primary studies are conducted, which resulted in making seven critical contributions to knowledge, as discussed in section 9.2 and the respective chapters. Finally, several opportunities for future work have been identified in section 9.3 to continue examining, enriching and resolving any new challenges arises to develop a suitable HAR approach for AAL system. In particular, evaluating and comparing the proposed single- and multi-user AR framework related studies with benchmark datasets collected from real-time multimodal sensing environment. As the ageing population continues to rise, the demand for AAL services in private home environments and professional services will increase to deliver realistic, responsive, and context-aware AAL applications. Therefore, it is anticipated that future research on AAL applications will benefit from the approaches, methods, and algorithms developed and evaluated in this thesis.

APPENDICES

The software package and user manual can be made available on request for interested readers.

REFERENCES

[1] R. Sterritt and C. Nugent, "Autonomic Computing and Ambient Assisted Living - Extended Abstract," *Eng. Auton. Auton. Syst. (EASe), 2010 Seventh IEEE Int. Conf. Work.*, pp. 149–151, 2010.

[2] X. Zhang, H. Wang, and Z. Yu, "Toward a Smart Home Environment for Elder People Based on Situation Analysis," *2010 7th Int. Conf. Ubiquitous Intell. Comput. 7th Int. Conf. Auton. Trust. Comput.*, pp. 7–12, 2010.

[3] C. Bohme, G. and Shingleton, "Invasive Technification: Critical Essays In The Philosophy Of Technology," in *Invasive Technification: Critical Essays In The Philosophy Of Technology*, Bloomsbury Publishing, 2012, p. 157.

[4] B. J. Banks, "The 'Age' of Opportunity," *IEEE Pulse*, vol. 8, no. 2, pp. 12–15, 2017.

[5] S. Zolfaghari, R. Zall, and M. R. Keyvanpour, "SOnAr: Smart Ontology Activity recognition framework to fulfill Semantic Web in smart homes Samaneh," in *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2016, vol. 2016-March, pp. 3339–3348.

[6] C. Debes, A. Merentitis, S. Sukhanov, M. Niessen, N. Frangiadakis, and A. Bauer, "Monitoring Activities of Daily Living in Smart Homes," *IEEE Signal Process. Mag.*, no. March, pp. 81–94, 2016.

[7] Z. Tang, J. Guo, S. Miao, S. Acharya, and J. H. Feng, "Ambient intelligence based context-aware assistive system to improve independence for people with autism spectrum disorder," *Proc. Annu. Hawaii Int. Conf. Syst. Sci.*, vol. 2016-March, pp. 3339–3348, 2016.

[8] M. Philipose *et al.*, "Inferring activities from interactions with objects," *IEEE Pervasive Comput.*, vol. 3, pp. 50–57, 2004.

[9] A. Hofmann and V. Davis, *Occupational Therapy Practice Framework: Domain and Process*, 2nd ed. AOTA Press, 2008.

[10] D. Sánchez, M. Tentori, and J. Favela, "Activity recognition for the smart hospital," *IEEE Intell. Syst.*, vol. 23, pp. 50–57, 2008.

[11] K. P. Murphy, “Dynamic Bayesian Networks: Representation, Inference and Learning,” 2002.

[12] Z. GHAHRAMANI, “AN INTRODUCTION TO HIDDEN MARKOV MODELS AND BAYESIAN NETWORKS,” *Int. J. Pattern Recognit. Artif. Intell.*, vol. 15, no. 01, pp. 9–42, 2001.

[13] J. Hong, E. Suh, and S.-J. Kim, “Context-aware systems: A literature review and classification,” *Expert Syst. Appl.*, vol. 36, no. 4, pp. 8509–8522, 2009.

[14] O. Brdiczka, J. L. Crowley, and P. Reignier, “Learning situation models in a smart home,” *IEEE Trans. Syst. Man, Cybern. Part B Cybern.*, vol. 39, no. 1, pp. 56–63, 2009.

[15] T. Nef *et al.*, “Evaluation of Three State-of-the-Art Classifiers for Recognition of Activities of Daily Living from Smart Home Ambient Data.,” *Sensors (Basel).*, vol. 15, no. 5, pp. 11725–40, 2015.

[16] R. Kumar, I. Qamar, J. S. Virdi, and N. C. Krishnan, “Multi-label Learning for Activity Recognition,” *2015 Int. Conf. Intell. Environ.*, pp. 152–155, 2015.

[17] M. Perkowitz, M. Philipose, K. Fishkin, and D. Patterson, “Mining models of human activities from the web,” *Proc. 13th Int. Conf. World Wide Web*, pp. 573–582, 2004.

[18] L. Chen, C. D. Nugent, and H. Wang, “A knowledge-driven approach to activity recognition in smart homes,” *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 6, pp. 961–974, 2012.

[19] D. Cook, K. Feuz, and N. Krishnan, “Transfer Learning for Activity Recognition: A Survey,” *Knowl. Inf. Syst.*, vol. 36, no. 3, pp. 537--556, 2013.

[20] L. Chen, C. Nugent, and G. Okeyo, “An ontology-based hybrid approach to activity modeling for smart homes,” *IEEE Trans. Human-Machine Syst.*, vol. 44, no. 1, pp. 92–105, 2014.

[21] T. Wu, C. Lian, and J. Y. Y. Hsu, “Joint recognition of multiple concurrent activities using factorial conditional random fields,” *Proc. 22nd Conf. Artif. Intell.*, pp. 82–87, 2007.

[22] D. H. Hu and Q. Yang, “CIGAR : Concurrent and Interleaving Goal and Activity Recognition,” *AAAI Conf. Artif. Intell.*, pp. 1363–1368, 2008.

[23] N. Nguyen and S. Venkatesh, “Recognising Behaviours of Multiple People with Hierarchial Probabilistic Model and Statistical Data Association,” *Br. Mach. Vis. Conf.*, pp. 1–10, 2006.

[24] G. Okeyo, L. Chen, and H. Wang, “Combining ontological and temporal formalisms for

composite activity modelling and recognition in smart homes," *Futur. Gener. Comput. Syst.*, vol. 39, pp. 29–43, 2014.

[25] J. Ye, G. Stevenson, and S. Dobson, "KCAR: A knowledge-driven approach for concurrent activity recognition," *Pervasive Mob. Comput.*, vol. 19, pp. 47–70, 2014.

[26] T. . Gu, L. . Wang, Z. . Wu, X. . Tao, and J. . Lu, "A pattern mining approach to sensor-based human activity recognition," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 9, pp. 1359–1372, 2011.

[27] G. Okeyo, L. Chen, H. Wang, and R. Sterritt, "Dynamic sensor data segmentation for real-time knowledge-driven activity recognition," *Pervasive Mob. Comput.*, vol. 10, pp. 155–172, 2014.

[28] M. Prossegger and A. Bouchachia, "Multi-resident Activity Recognition Using Incremental Decision Trees," in *Adaptive and Intelligent Systems*, Bouchachia., Springer International Publishing, 2014, pp. 182--191.

[29] A. Benmansour, A. Bouchachia, and M. Feham, "Multioccupant Activity Recognition in Pervasive Smart Home Environments," *ACM Comput. Surv.*, vol. 48, no. 3, pp. 1–36, 2015.

[30] C. Brayne and B. Miller, "Dementia and aging populations—A global priority for contextualized research and health policy," *PLOS Med.*, vol. 14, no. 3, p. e1002275, 2017.

[31] S. Duong, T. Patel, and F. Chang, "Dementia: What pharmacists need to know," *Can. Pharm. J.*, vol. 150, no. 2, pp. 118–129, 2017.

[32] C. Ye, Y. Xia, Y. Sun, S. Wang, H. Yan, and R. Mehmood, "ERAR: An Event-Driven Approach for Real-Time Activity Recognition," *2015 Int. Conf. Identification, Information, Knowl. Internet Things*, pp. 288–293, 2015.

[33] M. Cornacchia and Y. Zheng, "A Survey on Activity Detection and Classification Using Wearable Sensors," *IEEE Sens. J.*, vol. 17, no. 2, pp. 386–403, 2017.

[34] L. Chen, G. Okeyo, H. Wang, R. Sterritt, and C. Nugent, "A systematic approach to adaptive activity modeling and discovery in smart homes," *Proc. - 2011 4th Int. Conf. Biomed. Eng. Informatics, BMEI 2011*, vol. 4, pp. 2192–2196, 2011.

[35] L. Chen, C. Nugent, and J. Rafferty, "Ontology-based Activity Recognition Framework and Services," in *Proceedings of International Conference on Information Integration and Web-based Applications & Services - IIWAS '13*, 2013, pp. 463–469.

[36] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu, "Sensor-based activity

recognition," *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 42, no. 6, pp. 790–808, 2012.

[37] L. Liu *et al.*, "A framework of mining semantic-based probabilistic event relations for complex activity recognition," *Inf. Sci. (Ny).*, vol. 418–419, pp. 13–33, 2017.

[38] E. M. D. Jean-Baptiste and A. Mihailidis, "Analysis and comparison of two task models in a partially observable Markov decision process based assistive system," *2017 IEEE 4th Int. Conf. Soft Comput. Mach. Intell.*, pp. 183–187, 2017.

[39] Q. Ding, J. Han, and X. Zhao, "Continuous Estimation of Human Multi-Joint Angles from sEMG Using a State-Space Model," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 9, pp. 1518–1528, 2017.

[40] L. G. Fahad and M. Rajarajan, "Integration of discriminative and generative models for activity recognition in smart homes," *Appl. Soft Comput.*, vol. 37, pp. 992–1001, 2015.

[41] H. Bui, D. Phung, and S. Venkatesh, "Hierarchical hidden Markov models with general state hierarchy," *AAAI'04 Proc. 19th Natl. Conf. Artifical Intell.*, no. July, pp. 324–329, 2004.

[42] Y.-S. Lee and S.-B. Cho, "Hybrid Artificial Intelligent Systems: 6th International Conference, HAIS 2011, Wroclaw, Poland, May 23-25, 2011, Proceedings, Part I," E. Corchado, M. Kurzy\'nski, and M. Woźniak, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 460–467.

[43] The Pennsylvania State University, "Probability Theory and Mathematical Statistics." [Online]. Available: https://newonlinecourses.science.psu.edu/stat414/node/43/. [Accessed: 05-Sep-2019].

[44] G. Suresh kumar and G. Zayaraz, "Concept relation extraction using Naïve Bayes classifier for ontology-based question answering systems," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 27, no. 1, pp. 13–24, 2015.

[45] Y. Yemini, "Chapter 4 : Hidden Markov Models 4.1 Introduction to HMM," Columbia University, 2007.

[46] S. Saguna, A. Zaslavsky, and D. Chakraborty, "Complex activity recognition using context-driven activity theory and activity signatures," *ACM Trans. Comput. Interact.*, vol. 20, no. 6, pp. 1–34, 2013.

[47] S. Xing, T. Hanghang, and J. Ping, "Activity Recognition with Smartphone Sensors," *Consum. Commun. Netw. Conf. (CCNC), 2013 IEEE*, vol. 19, no. 3, pp. 235–249, 2014.

[48] L. Bao and S. S. Intille, "Activity Recognition from User-Annotated Acceleration Data,"

Pervasive Comput., pp. 1–17, 2004.

[49] D. Maguire and R. Frisby, "Comparison of Feature Classification Algorithms for Activity Recognition Based on Accelerometer and Heart Rate Data," *9th. IT T Conf.*, 2009.

[50] Y. Wang *et al.*, "A framework of energy efficient mobile sensing for automatic user state recognition," *MobiSys*, pp. 179–192, 2009.

[51] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical Machine Learning Tools and Techniques*. 2016.

[52] L. G. Fahad, A. Ali, and M. Rajarajan, "Information Science and Applications," J. K. Kim, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 819–826.

[53] D. M. J. Tax and R. P. W. Duin, "Support Vector Data Description," *Mach. Learn.*, vol. 54, no. 1, pp. 45–66, 2004.

[54] H. Guang-Xin, H.-F. Chen, and F. Yin, "IMPROVED SUPPORT VECTOR DATA DESCRIPTION," *Mach. Learn. Cybern.*, vol. 3, no. 10, pp. 1459–1463, 2010.

[55] V. Lai, D. Nguyen, K. Nguyen, and T. Le, "Mixture of Support Vector Data Descriptions," *Inf. Comput. Sci.*, pp. 135–140, 2015.

[56] J. H. Shin, B. Lee, and K. S. Park, "Detection of abnormal living patterns for elderly living alone using support vector data description.," *IEEE Trans. Inf. Technol. Biomed.*, vol. 15, no. 3, pp. 438–448, 2011.

[57] J. Ye *et al.*, "Semantic web technologies in pervasive computing: A survey and research roadmap," *Pervasive Mob. Comput.*, vol. 23, pp. 543–549, 2015.

[58] D. Wyatt, M. Philipose, and T. Choudhury, "Unsupervised activity recognition using automatically mined common sense," *Sensors*, vol. 20, no. 1, pp. 21–27, 2005.

[59] E. M. Tapia, T. Choudhury, and M. Philipose, "Building reliable activity models using hierarchical shrinkage and mined ontology," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3968 LNCS, pp. 17–32, 2006.

[60] P. Palmes, H. K. Pung, T. Gu, W. Xue, and S. Chen, "Object relevance weight pattern mining for activity recognition and segmentation," *Pervasive Mob. Comput.*, vol. 6, no. 1, pp. 43–57, 2010.

[61] L. Chen, C. Nugent, M. Mulvenna, D. Finlay, X. Hong, and M. Poland, "A Logical Framework for Behaviour Reasoning and Assistance in a Smart Home," *Int. J. Assist. Robot. Mechatronics*, vol. 9, no. 4, pp. 20–34, 2008.

[62] S. Staab and S. Rudi, *Handbook on Ontologies*, 2nd ed. Springer-Verlag Berlin

Heidelberg, 2009.

[63] B. N. Grosof, I. R. Horrocks, R. Volz, and S. Decker, “Description Logic Programs : Combining Logic Programs with Description Logic,” *WWW ’03 Proc. 12th Int. Conf. World Wide Web*, pp. 48–57, 2003.

[64] H. Hou and J. Wu, “Quasi-classical semantics and tableau calculus of description logics for paraconsistent reasoning in the semantic web,” *Proc. - 12th IEEE Int. Conf. Comput. Sci. Eng. CSE 2009*, vol. 2, pp. 703–708, 2009.

[65] M. Krotzsch, P. Hitzler, D. Vrandecic, and M. Sintek, “How to reason with OWL in a logic programming system,” *J. Int. Account. Res.*, vol. 7, no. 2, pp. 17–28, 2006.

[66] “What’s the difference between Tbox and Abox? - ANSWERS.” [Online]. Available: http://answers.semanticweb.com/questions/11802/whats-the-difference-between-tbox-and-abox. [Accessed: 03-Mar-2016].

[67] I. Viktoratos, A. Tsadiras, and N. Bassiliades, “A context-aware web-mapping system for group-targeted offers using semantic technologies,” *Expert Syst. Appl.*, vol. 42, no. 9, p. , 2015.

[68] M. Gavriushenko, M. Kankaanranta, and P. Neittaanmaki, “Semantically enhanced decision support for learning management systems,” *Proc. 2015 IEEE 9th Int. Conf. Semant. Comput. (IEEE ICSC 2015)*, pp. 298–305, 2015.

[69] K. L. Skillen, L. Chen, C. D. Nugent, M. P. Donnelly, W. Burns, and I. Solheim, “Ontological user modelling and semantic rule-based reasoning for personalisation of Help-On-Demand services in pervasive environments,” *Futur. Gener. Comput. Syst.*, vol. 34, pp. 97–109, 2014.

[70] D. Gaevic, D. Djuric, and V. Devedic, *Model Driven Architecture and Ontology Development | Dragan Gaevic | Springer*. Springer-Verlag Berlin Heidelberg, 2006.

[71] A. Ameen and B. P. Rani, “Extracting knowledge from Ontology using Jena for Semantic Web,” pp. 2–6, 2014.

[72] L. Liu, Y. Peng, M. Liu, and Z. Huang, “Sensor-based human activity recognition system with a multilayered model using time series shapelets,” *Knowledge-Based Syst.*, vol. 90, pp. 138–152, 2015.

[73] L. Chen, C. Nugent, S. Road, and N. Ireland, “Ontology-based Activity Recognition in Intelligent Pervasive Environments,” *Int. J. Web Inf. Syst.*, p. 30, 2009.

[74] L. S. Belisário and H. Pierreval, “Using genetic programming and simulation to learn how to dynamically adapt the number of cards in reactive pull systems,” *Expert Syst.*

Appl., vol. 42, no. 6, pp. 3129–3141, 2015.

[75] W. Pawgasame, "A Survey in Adaptive Hybrid Wireless Sensor Network for Military Operations," in *Defence Technology (ACDT)*, 2016, pp. 78–83.

[76] X. Hu, L. Yang, and W. Xiong, "A Novel Wireless Sensor Network Frame for Urban Transportation," *IEEE Internet Things J.*, vol. 2, no. 6, pp. 586–595, 2015.

[77] P. P. Gaikwad, J. P. Gabhane, and S. S. Golait, "A survey based on Smart Homes system using Internet-of-Things," in *2015 International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC)*, 2015, pp. 0330–0335.

[78] I. Khan, F. Belqasmi, R. Glitho, N. Crespi, M. Morrow, and P. Polakos, "Wireless Sensor Network Virtualization: A Survey," *IEEE Commun. Surv. Tutorials*, vol. 18, no. 1, pp. 553–576, 2016.

[79] M. Ziaeefar and R. Bergevin, "Semantic human activity recognition: A literature review," *Pattern Recognit.*, vol. 48, no. 8, pp. 2329–2345, 2015.

[80] Quora, "Image Processing: Difference between SURF and SIFT?where and when to use this algo?," 2014. [Online]. Available: https://www.quora.com/Image-Processing/Difference-between-SURF-and-SIFT-where-and-when-to-use-this-algo.

[81] N. Hamid, A. Yahya, R. B. Ahmad, and O. M. Al-qershi, "A Comparison between Using SIFT and SURF for Characteristic Region Based Image Steganography," vol. 9, no. 3, pp. 110–116, 2012.

[82] S. Govindarajulu and K. N. K. Reddy, "A Comparison of SIFT , PCA-SIFT and SURF," vol. 1, no. 3, pp. 53–60, 2014.

[83] G. M. Zafaruddin and H. S. Fadewar, "Face recognition: A holistic approach review," *Proc. 2014 Int. Conf. Contemp. Comput. Informatics, IC3I 2014*, pp. 175–178, 2014.

[84] W. Zhang, Y. Zhang, L. Ma, J. Guan, and S. Gong, "Multimodal learning for facial expression recognition," *Pattern Recognit.*, vol. 48, no. 10, pp. 1–12, 2015.

[85] N. N. Khatri, Z. H. Shah, and S. a Patel, "Facial Expression Recognition : A Survey," *Procedia - Procedia Comput. Sci.*, vol. 5, no. 1, pp. 149–152, 2014.

[86] J. K. Aggarwal and L. Xia, "Human activity recognition from 3D data: A review," *Pattern Recognit. Lett.*, vol. 48, pp. 70–80, 2014.

[87] R. Fallahzadeh, M. Pedram, R. Saeedi, B. Sadeghi, M. Ong, and H. Ghasemzadeh, "Smart-Cuff: A wearable bio-sensing platform with activity-sensitive information quality assessment for monitoring ankle edema," *2015 IEEE Int. Conf. Pervasive Comput. Commun. Work. PerCom Work. 2015*, pp. 57–62, 2015.

[88] Karen Lightman, “Next-Gen Sensors Make Golf Clubs, Tennis Rackets, and Baseball Bats Smarter Than Ever - IEEE Spectrum,” *IEEE Spectrum*, 2016. [Online]. Available: http://spectrum.ieee.org/consumer-electronics/gadgets/nextgen-sensors-make-golf-clubs-tennis-rackets-and-baseball-bats-smarter-than-ever. [Accessed: 02-Mar-2016].

[89] S. Kosir, “A Look At Smart Clothing For 2015 | Wearable Technologies,” 23-Mar-2015. [Online]. Available: https://www.wearable-technologies.com/2015/03/a-look-at-smartclothing-for-2015/. [Accessed: 03-Mar-2016].

[90] D. J. Cassar and M. a. Saliba, “A Force Feedback Glove Based on Magnetorheological Fluid: Prototype Development and Evaluation,” *Int. Conf. Appl. Biomech. Bionics*, pp. 618–623, 2010.

[91] E. Ackerman, “Hands-On with Ultrahaptics’ Invisible, Touchable Controls - IEEE Spectrum,” *IEEE Spectrum*, 2016. [Online]. Available: http://spectrum.ieee.org/tech-talk/consumer-electronics/gadgets/handson-with-ultrahaptics-invisible-touchable-controls. [Accessed: 02-Mar-2016].

[92] C. Chui, “Blood Glucose Regulation using an Implantable Artificial Pancreas,” p. 8221, 2015.

[93] C. Occhiuzzi and G. Marrocco, “Human body sensing: A pervasive approach by implanted RFID tags,” *2010 3rd Int. Symp. Appl. Sci. Biomed. Commun. Technol. ISABEL 2010*, no. 2, 2010.

[94] R. Lodato, V. Lopresto, R. Pinto, and G. Marrocco, “Numerical and experimental characterization of through-the-body UHF-RFID links for passive tags implanted into human limbs,” *IEEE Trans. Antennas Propag.*, vol. 62, no. 10, pp. 5298–5306, 2014.

[95] G. Imai, K. Matsuda, H. Takahata, and M. Okada, “Particle filter-assisted positioning method for identifying RFID-tag implanted in the organism,” *Middle East Conf. Biomed. Eng. MECBME*, pp. 313–316, 2014.

[96] J.-L. Reyes-Ortiz, L. Oneto, A. Samà, X. Parra, and D. Anguita, “Transition-Aware Human Activity Recognition Using Smartphones,” *Neurocomputing*, vol. 171, pp. 754–767, 2015.

[97] H. Wang, J. Saboune, and A. El Saddik, “Control your smart home with an autonomously mobile smartphone,” *Electron. Proc. 2013 IEEE Int. Conf. Multimed. Expo Work. ICMEW 2013*, 2013.

[98] M. Shoaib, H. Scholten, and P. J. M. Havinga, “Towards Physical Activity Recognition Using Smartphone Sensors,” *2013 IEEE 10th Int. Conf. Ubiquitous Intell. Comput. 2013 IEEE 10th Int. Conf. Auton. Trust. Comput.*, vol. 19, no. 3, pp. 80–87, 2013.

[99] D. Mitchell, P. J. Morrow, and C. D. Nugent, "A Sensor and Video Based Ontology for Activity Recognition in Smart Environments," *IEEE Eng. Med. Biol. Soc.*, pp. 5932–5935, 2014.

[100] E. Shailendra and P. K. Bhatia, "Analyzing Home Automation and Networking Technologies," *IEEE Potentials*, vol. 37, no. 1, pp. 27–33, 2018.

[101] T. Malche and P. Maheshwary, "Internet of Things (IoT) for building Smart Home System," in *IoT in Social, Mobile, Analytics and Cloud*, 2017, pp. 65–70.

[102] A. Nayyar, "Real Time Smart Home Automation based on PIC Microcontroller, Bluetooth and Android Technology," *2016 3rd Int. Conf. Comput. Sustain. Glob. Dev.*, pp. 1478–1484, 2016.

[103] H. V. Bhatnagar, P. Kumar, S. Rawat, and T. Choudhury, "Implementation model of Wi-Fi based Smart Home System," *Proc. 2018 Int. Conf. Adv. Comput. Commun. Eng. ICACCE 2018*, no. June, pp. 23–28, 2018.

[104] S. Sasirekha, I. J. L. Paul, and S. Swamynathan, "An API Centric Smart Kitchen Application," in *2nd International Conference on Computer, Communication, and Signal Processing: Special Focus on Technology and Innovation for Smart Environment, ICCCSP 2018*, 2018, no. Icccsp, pp. 1–6.

[105] V. Govindraj, M. Sathiyanarayanan, and B. Abubakar, "Customary homes to smart homes using Internet of Things (IoT) and mobile application," *Proc. 2017 Int. Conf. Smart Technol. Smart Nation, SmartTechCon 2017*, pp. 1059–1063, 2018.

[106] P. Nath and U. C. Pati, "Low-Cost Android App Based Voice Operated Room Automation System," in *2018 3rd International Conference for Convergence in Technology (I2CT)*, 2018, pp. 1–4.

[107] P. Kumar and U. C. Pati, "Arduino and Raspberry Pi based smart communication and control of home appliance system," *Proc. 2016 Online Int. Conf. Green Eng. Technol. IC-GET 2016*, pp. 1–6, 2017.

[108] M. Espinilla, L. Martinez, J. Medina, and C. Nugent, "The experience of developing the UJAmI Smart Lab," *IEEE Access*, vol. 6, pp. 34631–34642, 2018.

[109] Daniel Roggen, Alberto Calatroni, Long-Van Nguyen-Dinh, Ricardo Chavarriaga, Hesam Sagha, and Sundara Tejaswi Digumarti, "UCI Machine Learning Repository: OPPORTUNITY Activity Recognition Data Set," 2012. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/OPPORTUNITY+Activity+Recognition. [Accessed: 24-Feb-2019].

[110] J. Rafferty *et al.*, "A scalable, research oriented, generic, sensor data platform," *IEEE*

Access, vol. 6, pp. 45473–45484, 2018.

[111] A. G. S. Comas *et al.*, "H2Al—The Human Health and Activity Laboratory," *Proceedings*, vol. 2, no. 19, p. 1241, 2018.

[112] S. Dernbach, B. Das, N. C. Krishnan, B. L. Thomas, and D. J. Cook, "Simple and Complex Activity Recognition through Smart Phones," pp. 214–221, 2012.

[113] G. Ferroni, R. Bonfigli, E. Principi, S. Squartini, and F. Piazza, "A Deep Neural Network Approach for Voice Activity Detection in Multi-Room Domestic Scenarios," pp. 1–8, 2015.

[114] Y. Liu, L. Nie, L. Liu, and D. S. Rosenblum, "From Action to Activity: Sensor-Based Activity Recognition," *Neurocomputing*, pp. 1–8, 2015.

[115] K. S. Gayathri, S. Elias, and S. Shivashankar, *Composite Activity Recognition in Smart Homes Using Markov Logic Network*. 2015.

[116] D. Riboni and C. Bettini, "OWL 2 modeling and reasoning with complex human activities," *Pervasive Mob. Comput.*, vol. 7, no. 3, pp. 379–395, 2011.

[117] X. Hu, T. H. S. Chu, V. C. M. Leung, E. C.-H. Ngai, P. Kruchten, and H. C. B. Chan, "A Survey on Mobile Social Networks: Applications, Platforms, System Architectures, and Future Research Directions," *Commun. Surv. Tutorials, IEEE*, vol. PP, no. 99, p. 1, 2014.

[118] G. E. Modoni, M. Sacco, and W. Terkaj, "A survey of RDF store solutions," *2014 Int. Conf. Eng. Technol. Innov.*, no. January, pp. 1–7, 2014.

[119] J. A. Botia, A. Villa, and J. Palma, "Ambient Assisted Living system for in-home monitoring of healthy independent elders," *Expert Syst. Appl.*, vol. 39, no. 9, pp. 8136–8148, 2012.

[120] G. Meditskos, S. Dasiopoulou, and I. Kompatsiaris, "MetaQ: A knowledge-driven framework for context-aware activity recognition combining SPARQL and OWL 2 activity patterns," *Pervasive Mob. Comput.*, 2015.

[121] I.-H. Bae, "An ontology-based approach to ADL recognition in smart homes," *Futur. Gener. Comput. Syst.*, vol. 33, pp. 32–41, 2014.

[122] Amazon Developer, "Alexa - Build engaging voice experiences for your services and devices." [Online]. Available: https://developer.amazon.com/public/solutions/alexa. [Accessed: 01-May-2016].

[123] Samsung, "SmartThings Hub | SAMSUNG UK," 2016. [Online]. Available: https://www.samsung.com/uk/apps/smartthings/. [Accessed: 08-Jun-2019].

[124] IFTTT, "Recipes on IFTTT are the easy way to automate your world," *IFTTT*. [Online]. Available: https://ifttt.com/. [Accessed: 20-May-2016].

[125] M. S. Perez and E. Carrera, "Time synchronization in Arduino-based wireless sensor networks," *IEEE Lat. Am. Trans.*, vol. 13, no. 2, pp. 455–461, 2015.

[126] Samsung SmartThings, "SmartThings Shield for Arduino," *shop.smartthings.com*. [Online]. Available: https://shop.smartthings.com/#!/products/smartthings-shield-arduino. [Accessed: 10-Jun-2016].

[127] G. Civitarese, T. Sztyler, D. Riboni, C. Bettini, and H. Stuckenschmidt, "POLARIS: Probabilistic and Ontological Activity Recognition in Smart-homes," *IEEE Trans. Knowl. Data Eng.*, vol. PP, no. c, pp. 1–1, 2019.

[128] D. N. Mekuria, P. Sernani, N. Falcionelli, and A. F. Drazoni, "A Probabilistic Multi-Agent System Architecture for Reasoning in Smart Homes," in *2019 IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA)*, 2019, pp. 1–6.

[129] A. Gaur, B. Scotney, G. Parr, and S. McClean, "COSA: An approach to classifying uncertainty in an ontology tree for sensor data Fusion," in *SAS 2016 - Sensors Applications Symposium, Proceedings*, 2016, pp. 320–325.

[130] X. Hong, C. Nugent, M. Mulvenna, S. McClean, B. Scotney, and S. Devlin, "Evidential fusion of sensor data for activity recognition in smart homes," *Pervasive Mob. Comput.*, vol. 5, no. 3, pp. 236–252, 2009.

[131] C. Langensiepen and A. Lotfi, "Uncertainty measures in an ambient intelligence environment," in *IEEE International Conference on Fuzzy Systems*, 2017, pp. 0–4.

[132] L. Pavlič, M. Hericko, V. Podgorelec, and I. Rozman, "Improving design pattern adoption with an ontology-based repository," *Inform.*, 2009.

[133] M. Ali and M. O. Elish, "A Comparative Literature Survey of Design Patterns Impact on Software Quality," *Inf. Sci. Appl. (ICISA), 2013 Int. Conf.*, pp. 1–7, 2013.

[134] C. Zhang, D. Budgen, and S. Drummond, "Using a follow-on survey to investigate why use of the visitor, singleton & facade patterns is controversial," in *Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement - ESEM '12*, 2012, p. 79.

[135] J. Davies, D. Fensel, and F. Van Harmelen, *Towards the Semantic Web: Ontology-Driven Knowledge Management*. 2002.

[136] Apache, "Apache Jena," *Apache*. [Online]. Available: https://jena.apache.org/.

[Accessed: 19-Jun-2015].

[137] S. Powers, *Practical RDF*, 1st ed. O'Reilly Media, 2003.

[138] W3C, "OWL 2 Web Ontology Language Document Overview (Second Edition)," *W3C*, 2012. [Online]. Available: https://www.w3.org/TR/owl2-overview/. [Accessed: 05-Jan-2016].

[139] B. Ducbarme, *Learning SPARQL*, 1st ed. O'Reilly Media, 2013.

[140] J. Wan, M. J. O'Grady, and G. M. P. O'Hare, "Dynamic sensor event segmentation for real-time activity recognition in a smart home context," *Pers. Ubiquitous Comput.*, vol. 19, no. 2, pp. 287–301, 2015.

[141] D. R. Faria, M. Vieira, C. Premebida, and U. Nunes, "Probabilistic Human Daily Activity Recognition towards Robot-assisted Living," pp. 582–587, 2015.

[142] Y. Zhong, "A theory of semantic information," *China Commun.*, vol. 14, no. 1, pp. 1–17, 2017.

[143] A. Tarski, "The Semantic Conception of Truth: and the Foundations of Semantics," *Philosophy and Phenomenological Research*, vol. 4, no. 3. p. 341, 1944.

[144] J. Rafferty, C. D. Nugent, J. Liu, and L. Chen, "From Activity Recognition to Intention Recognition for Assisted Living Within Smart Homes," *IEEE Trans. Human-Machine Syst.*, vol. PP, no. 99, pp. 1–12, 2016.

[145] E. Della Valle and M. Grossniklaus, "C-SPARQL: A CONTINUOUS QUERY LANGUAGE FOR RDF DATA STREAMS," *Int. J. Semant. Comput.*, vol. 04, no. 01, pp. 3–25, 2010.

[146] G. Okeyo, L. Chen, H. Wang, and R. Sterritt, "A hybrid ontological and temporal approach for composite activity modelling," *Proc. 11th IEEE Int. Conf. Trust. Secur. Priv. Comput. Commun. Trust. - 11th IEEE Int. Conf. Ubiquitous Comput. Commun. IUCC-2012*, pp. 1763–1770, 2012.

[147] R. Culmone, P. Giuliodori, and M. Quadrini, "Human Activity Recognition using a Semantic Ontology-Based Framework," *Int. J. Adv. Intell. Syst.*, vol. 8, no. 2, pp. 159–168, 2015.

[148] U. Naeem, "Activities of daily life recognition using process representation modelling to support intention analysis," *Int. J. Pervasive Comput. Commun.*, vol. 11, no. 3, p. 347, 2015.

[149] X. Hong and C. D. Nugent, "Segmenting sensor data for activity monitoring in smart environments," *Pers. Ubiquitous Comput.*, vol. 17, no. 3, pp. 545–559, 2013.

[150] P. Vickers, "Understanding Visualisation: A Formal Foundation using Category Theory and Semiotics," *IEEE Trans. Vis. Comput. Graph.*, vol. X, no. X, pp. 1–14, 2013.

[151] Y. Wang, "Formal rules for concept and semantics manipulations in cognitive linguistics and machine learning," in *2017 IEEE 16th International Conference on Cognitive Informatics Cognitive Computing (ICCI*CC)*, 2017, pp. 43–50.

[152] D. Triboan, L. Chen, F. Chen, S. Fallmann, and I. Psychoula, "Real-Time Sensor Observation Segmentation For Complex Activity Recognition Within Smart Environments," in *2017 IEEE 14th International Conference on Ubiquitous Intelligence and Computing (UIC 2017)*, 2017.

[153] D. Triboan, L. Chen, F. Chen, and Z. Wang, "Semantic segmentation of real-time sensor data stream for complex activity recognition," *Pers. Ubiquitous Comput.*, vol. 21, no. 3, pp. 411–425, 2017.

[154] S. Abburu, "A Survey on Ontology Reasoners and Comparison," *Int. J. Comput. Appl.*, vol. 57, no. 17, pp. 33–39, 2012.

[155] K. Dentler, R. Cornet, A. Ten Teije, and N. De Keizer, "Comparison of reasoners for large ontologies in the OWL 2 EL profile," *Semant. Web*, vol. 2, no. 2, pp. 71–87, 2011.

[156] G. De Giacomo and M. Lenzerini, "TBox and ABox Reasoning in Expressive Description Logics," *Proc. Fifth Int. Conf. Princ. Knowl. Represent. Reason.*, pp. 316–327, 1996.

[157] D. Triboan, L. Chen, F. Chen, and Z. Wang, "Towards a Service-Oriented Architecture for a Mobile Assistive System with Real-time Environmental Sensing," *TSINGHUA Sci. Technol.*, vol. 21, no. 6, pp. 581–597, 2016.

[158] D. Triboan, L. Chen, and F. Chen, "Towards a Mobile Assistive System Using Service-oriented Architecture," in *2016 IEEE Symposium on Service-Oriented System Engineering Towards*, 2016, vol. 21, no. 6, pp. 187–196.

[159] Stanford University, "Protégé." [Online]. Available: https://protege.stanford.edu/. [Accessed: 17-Jun-2018].

[160] R. Volz, S. Staab, and B. Motik, "Incremental Maintenance Of Materialized Ontologies," *Lect. Notes Comput. Sci.*, vol. 2888/2003, no. 2888/2003, pp. 707–724, 2003.

[161] W3C, "SPIN - Overview and Motivation," 2011. [Online]. Available: https://www.w3.org/Submission/spin-overview/. [Accessed: 22-Aug-2016].

[162] B. Cuenca Grau, C. Halaschek-Wiener, and Y. Kazakov, "History matters: Incremental

ontology reasoning using modules," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 4825 LNCS, pp. 183–196, 2007.

[163] Y. Ren, J. Z. Pan, I. Guclu, and M. Kollingbaum, "A combined approach to incremental reasoning for EL ontologies," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9898 LNCS, no. August, pp. 167–183, 2016.

[164] M. Peters, C. Brink, S. Sachweh, and A. Zündorf, "Scaling parallel rule-based reasoning," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8465 LNCS, pp. 270–285, 2014.

[165] J. Ponge, "Fork and Join: Java Can Excel at Painless Parallel Programming Too!," 2011. [Online]. Available: http://www.oracle.com/technetwork/articles/java/fork-join-422606.html. [Accessed: 18-Jun-2017].

[166] H. Medjahed, D. Istrate, J. Boudy, and B. Dorizzi, "Human activities of daily living recognition using fuzzy logic for elderly home monitoring," *2009 IEEE Int. Conf. Fuzzy Syst. August 20, 2009 - August 24, 2009*, vol. 33, no. 0, pp. 2001–2006, 2009.

[167] C. Lea, G. D. Hager, and R. Vidal, "An improved model for segmentation and recognition of fine-grained activities with application to surgical training tasks," in *Proceedings - 2015 IEEE Winter Conference on Applications of Computer Vision, WACV 2015*, 2015, pp. 1123–1129.

[168] N. Díaz Rodríguez, M. P. Cuéllar, J. Lilius, and M. Delgado Calvo-Flores, "A fuzzy ontology for semantic modelling and recognition of human behaviour," *Knowledge-Based Syst.*, vol. 66, pp. 46–60, 2014.

[169] M. H. M. Noor, Z. Salcic, and K. I. K. Wang, "Enhancing ontological reasoning with uncertainty handling for activity recognition," *Knowledge-Based Syst.*, vol. 114, pp. 47–60, 2016.

[170] D. De, P. Bharti, S. K. Das, and S. Chellappan, "Multimodal wearable sensing for fine-grained activity recognition in healthcare," *IEEE Internet Comput.*, vol. 19, no. 5, pp. 26–35, 2015.

[171] S. El-Sappagh and M. Elmogy, "A fuzzy ontology modeling for case base knowledge in diabetes mellitus domain," *Eng. Sci. Technol. an Int. J.*, vol. 20, no. 3, pp. 1025–1040, 2017.

[172] A. C. Bukhari and Y. G. Kim, "Integration of a secure type-2 fuzzy ontology with a multi-agent platform: A proposal to automate the personalized flight ticket booking domain," *Inf. Sci. (Ny).*, vol. 198, pp. 24–47, 2012.

[173] L. Liu, S. Wang, Y. Peng, Z. Huang, M. Liu, and B. Hu, “Mining intricate temporal rules for recognizing complex activities of daily living under uncertainty,” *Pattern Recognit.*, vol. 60, pp. 1015–1028, 2016.

[174] D. Ausín, D. López-De-Ipiña, and F. Castanedo, “A probabilistic OWL reasoner for intelligent environments,” *CEUR Workshop Proc.*, vol. 1259, no. Ursw, pp. 1–12, 2014.

[175] P. C. Roy, S. R. Abidi, and S. S. R. Abidi, “Possibilistic activity recognition with uncertain observations to support medication adherence in an assisted ambient living setting,” *Knowledge-Based Syst.*, vol. 133, pp. 156–173, 2017.

[176] K. Boutouhami, S. Benferhat, F. Khellaf, and F. Nouioua, “Uncertain lightweight ontologies in a product-based possibility theory framework,” *Int. J. Approx. Reason.*, vol. 88, pp. 237–258, 2017.

[177] H. Kim *et al.*, “Collaborative classification for daily activity recognition with a smartwatch,” in *2016 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2016 - Conference Proceedings*, 2017, pp. 3707–3712.

[178] A. Moschetti, L. Fiorini, D. Esposito, P. Dario, and F. Cavallo, “Daily activity recognition with inertial ring and bracelet: An unsupervised approach,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2017, pp. 3250–3255.

[179] G. Hu, X. Qiu, and L. Meng, “Human activity recognition based on Hidden Markov Models using computational RFID,” *2017 4th Int. Conf. Syst. Informatics, ICSAI 2017*, vol. 2018-Janua, no. Icsai, pp. 813–818, 2018.

[180] G. Civitarese, C. Bettini, T. Sztyler, D. Riboni, and H. Stuckenschmidt, “NECTAR : Knowledge-based Collaborative Active Learning for Activity Recognition,” in *2018 IEEE International Conference on Pervasive Computing and Communications : PerCom 2018, Athens, Greece, March 19-23, 2018*, 2018, pp. 125–134.

[181] M. A. Razzaq *et al.*, “mlCAF: Multi-level cross-domain semantic context fusioning for behavior identification,” *Sensors (Switzerland)*, vol. 17, no. 10, pp. 1–25, 2017.

[182] G. Meditskos and I. Kompatsiaris, “iKnow: Ontology-driven situational awareness for the recognition of activities of daily living,” *Pervasive Mob. Comput.*, vol. 40, pp. 17–41, 2017.

[183] H. Medjahed, D. Istrate, J. Boudy, J.-L. Baldinger, and B. Dorizzi, “A pervasive multi-sensor data fusion for smart home healthcare monitoring,” *2011 IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE 2011)*, pp. 1466–1473, 2011.

[184] J. Shell and S. Coupland, “Improved Decision Making Using Fuzzy Temporal Relationships within Intelligent Assisted Living Environments,” *2011 Seventh Int. Conf.*

Intell. Environ., pp. 149–156, 2011.

[185] N. Wichit, “Multisensor data fusion model for activity detection,” in *12th International Conference on ICT and Knowledge Engineering*, 2014, pp. 54–59.

[186] F. Bobillo and U. Straccia, “Fuzzy ontology representation using OWL 2,” *Int. J. Approx. Reason.*, vol. 52, no. 7, pp. 1073–1094, 2011.

[187] F. Bobillo and U. Straccia, “The fuzzy ontology reasoner fuzzyDL,” *Knowledge-Based Syst.*, vol. 95, pp. 12–34, 2016.

[188] V. V. Cross, “Fuzzy ontologies: The state of the art,” *2014 IEEE Conf. Norbert Wiener 21st Century Driv. Technol. Futur. 21CW 2014 - Inc. Proc. 2014 North Am. Fuzzy Inf. Process. Soc. Conf. NAFIPS 2014, Conf. Proc.*, 2014.

[189] D. Triboan, L. Chen, and F. Chen, “Fuzzy-based Fine-grained Human Activity Recognition within Smart Environments,” in *2019 IEEE Smart World Congress*, 2019, pp. 94–101.

[190] W3C, “Pellet - Semantic Web Standards,” *W3C*, 2011. [Online]. Available: https://www.w3.org/2001/sw/wiki/Pellet. [Accessed: 08-Jan-2015].

[191] W3C, “SPARQL,” *W3C*, 2013. [Online]. Available: https://www.w3.org/TR/sparql11-overview/. [Accessed: 08-Jan-2015].

[192] F. Bobillo and U. Straccia, “Generalizing type-2 fuzzy ontologies and type-2 fuzzy description logics,” *Int. J. Approx. Reason.*, vol. 87, pp. 40–66, 2017.

[193] Jersey and Oracle, “RESTful Web Services in Java.” [Online]. Available: https://jersey.github.io/. [Accessed: 08-Jan-2015].

[194] K. S. Gayathri, K. S. Easwarakumar, and S. Elias, “Probabilistic ontology based activity recognition in smart homes using Markov Logic Network,” *Knowledge-Based Syst.*, vol. 121, pp. 173–184, 2017.

[195] S. Ebrahimi, M. Elhoseiny, T. Darrell, and M. Rohrbach, “Uncertainty-guided Continual Learning with Bayesian Neural Networks,” 2019.

[196] I. Riali, M. Fareh, and H. Bouarfa, “A semantic approach for handling probabilistic knowledge of fuzzy ontologies,” in *ICEIS 2019 - Proceedings of the 21st International Conference on Enterprise Information Systems*, 2019, vol. 1, pp. 395–402.

[197] G. Kovács and K. M. Spens, “Abductive reasoning in logistics research,” *Int. J. Phys. Distrib. Logist. Manag.*, vol. 35, no. 2, pp. 132–144, 2005.

[198] L. Su, G. Li, and L. Zhao, “Dense Sensing-Based Composite Activity Recognition in Smart Home,” in *Proceedings - 2018 10th International Conference on Intelligent*

Human-Machine Systems and Cybernetics, IHMSC 2018, 2018, vol. 1, pp. 9–14.

[199] Z. Ding, Y. Peng, and R. Pan, “BayesOWL: Uncertainty modeling in semantic web ontologies,” *Stud. Fuzziness Soft Comput.*, vol. 204, pp. 3–29, 2006.

[200] Yi Yang and J. Calmet, “OntoBayes: An Ontology-Driven Uncertainty Model,” in *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC’06)*, 2005, vol. 1, pp. 457–463.

[201] D. Ausín, F. Castanedo, and D. López-de-Ipiña, “TURAMBAR: An approach to deal with uncertainty in semantic environments,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7657 LNCS, pp. 329–337, 2012.

[202] R. N. Carvalho, K. B. Laskey, and P. C. G. Costa, “PR-OWL 2.0 - Bridging the gap to OWL semantics Why map PR-OWL Random Variables to OWL,” *Proc. 6th Int. Work. Uncertain. Reason. Semant. Web (URSW 2010), collocated with 9th Int. Semant. Web Conf. (ISWC 2010)*, pp. 73–84, 2010.

[203] R. N. Carvalho, K. B. Laskey, and P. C. G. Costa, “PR-OWL – a language for defining probabilistic ontologies,” *Int. J. Approx. Reason.*, vol. 91, pp. 56–79, 2017.

[204] R. N. Carvalho, M. Ladeira, R. M. De Souza, S. Matsumoto, H. A. Da Rocha, and G. L. Mendes, “UMP-ST plug-in: A tool for documenting, maintaining, and evolving probabilistic ontologies,” in *CEUR Workshop Proceedings*, 2013, vol. 1073, pp. 15–26.

[205] R. N. Carvalho, M. Ladeira, and L. Weigang, “Probabilistic Ontologies Incremental Modeling Using UnBBayes,” 2011.

[206] R. N. Carvalho, K. B. Laskey, P. C. G. Costa, M. Ladeira, L. L. Santos, and S. Matsumoto, “Probabilistic ontology and knowledge fusion for procurement fraud detection in Brazil,” *CEUR Workshop Proc.*, vol. 527, pp. 3–14, 2009.

[207] C. Young and P. Student, “Multi-Entity Bayesian Networks Learning In Predictive Situation Awareness,” in *18TH INTERNATIONAL COMMAND AND CONTROL RESEARCH AND TECHNOLOGY SYMPOSIUM*, 2013, vol. 4444, no. 703, pp. 1894–1901.

[208] R. N. Carvalho, P. C. G. Costa, K. B. Laskey, and K. C. Chang, “PROGNOS: Predictive situational awareness with probabilistic ontologies,” in *2010 13th International Conference on Information Fusion*, 2010, pp. 1–8.

[209] E. Wright, S. Mahoney, K. Laskey, M. Takikawa, and T. Levitt, “Multi-entity Bayesian networks for situation assessment,” *Proc. 5th Int. Conf. Inf. Fusion, FUSION 2002*, vol.

2, pp. 804–811, 2002.

[210] S. R Patnaik Patnaikuni and S. R Gengaje, "Survey of Multi Entity Bayesian Networks (MEBN) and its applications in probabilistic reasoning," *Int. J. Adv. Res. Comput. Sci.*, vol. 8, no. 5, pp. 2425–2429, 2017.

[211] K. B. Laskey, "MEBN: A language for first-order Bayesian knowledge bases," *Artif. Intell.*, vol. 172, no. 2–3, pp. 140–178, 2008.

[212] K. B. Laskey, R. Haberlin, P. Costa, and R. N. Carvalho, "PR-OWL 2 case study: A maritime domain probabilistic ontology," *CEUR Workshop Proc.*, vol. 808, pp. 76–83, 2011.

[213] F. Setiawan, E. Budiardjo, and W. Wibowo, "ByNowLife: A Novel Framework for OWL and Bayesian Network Integration," *Information*, vol. 10, no. 3, p. 95, 2019.

[214] S. Matsumoto, R. Carvalho, and M. Ladeira, "UnBBayes: a java framework for probabilistic models in AI," *... Press. http//unbbayes. ...*, no. July 2015, 2011.

[215] D. Riboni, T. Sztyler, G. Civitarese, and H. Stuckenschmidt, "Unsupervised recognition of interleaved activities of daily living through ontological and probabilistic reasoning," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing - UbiComp '16*, 2016, pp. 1–12.

[216] H. Lee, J. S. Choi, and R. Elmasri, "Sensor data fusion using DSm theory for activity recognition under uncertainty in home-based care," in *Proceedings - International Conference on Advanced Information Networking and Applications, AINA*, 2009, pp. 517–524.

[217] A. Calzada, J. Liu, C. D. Nugent, H. Wang, and L. Martinez, "Sensor-based activity recognition using extended belief rule-based inference methodology," in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC 2014*, 2014, pp. 2694–2697.

[218] F. Sebbak, S. Bouznad, F. Benhammadi, A. Chibani, and Y. Amirat, "Context Awareness in Uncertain Pervasive Computing and Sensors Environment," in *2018 21st International Conference on Information Fusion, FUSION 2018*, 2018, pp. 1026–1031.

[219] G. Der Wu and Z. W. Zhu, "Fuzzy Neural Network with Audio-Visual Data for Voice Activity Detection in Noisy Environments," in *2018 International Conference on Intelligent Autonomous Systems, ICoIAS 2018*, 2018, pp. 141–145.

[220] A. Syafiq, A. Sukor, A. Zakaria, N. A. Rahim, and R. Setchi, "Semantic Knowledge Base in Support of Activity Recognition in Smart Home Environments," *Int. J. Eng. Technol.*, no. December, 2018.

[221] M. Compton *et al.*, “The SSN ontology of the W3C semantic sensor network incubator group,” *J. Web Semant.*, vol. 17, pp. 25–32, 2012.

[222] T. Di Noia, M. Mongiello, F. Nocera, and U. Straccia, “A fuzzy ontology-based approach for tool-supported decision making in architectural design,” *Knowl. Inf. Syst.*, vol. 58, no. 1, pp. 83–112, 2019.

[223] J. A. Morente-Molinera, G. Kou, R. González-Crespo, J. M. Corchado, and E. Herrera-Viedma, “Solving multi-criteria group decision making problems under environments with a high number of alternatives using fuzzy ontologies and multi-granular linguistic modelling methods,” *Knowledge-Based Syst.*, vol. 137, pp. 54–64, 2017.

[224] J. Wu, Y. Feng, and P. Sun, “Sensor fusion for recognition of activities of daily living,” *Sensors (Switzerland)*, vol. 18, no. 11, pp. 1–13, 2018.

[225] W3C, “Semantic Sensor Network Ontology,” *Semantic Web*, 19-Oct-2017. [Online]. Available: https://www.w3.org/TR/vocab-ssn/. [Accessed: 09-Nov-2018].

[226] Y.-T. Chiang, K.-C. Hsu, C.-H. Lu, L.-C. Fu, and J. Y.-J. Hsu, “Interaction models for multiple-resident activity recognition in a smart home,” *IEEE/RSJ 2010 Int. Conf. Intell. Robot. Syst. IROS 2010 - Conf. Proc.*, pp. 3753–3758, 2010.

[227] Y. Kim, J. An, M. Lee, and Y. Lee, “An Activity-Embedding Approach for Next-Activity Prediction in a Multi-User Smart Space,” 2017.

[228] D. Gordon, J.-H. Hanne, M. Berchtold, A. A. N. Shirehjini, and M. Beigl, “Towards Collaborative Group Activity Recognition Using Mobile Devices,” *Mob. Networks Appl.*, vol. 18, no. 3, pp. 326–340, 2012.

[229] A. Doryab and J. Togelius, “Activity Recognition in Collaborative Environments,” pp. 10–15, 2012.

[230] S. Schneegass, “SkullConduct : Biometric User Identification on Eyewear Computers Using Bone Conduction Through the Skull,” *Conf. Hum. Factors Comput. Syst.*, pp. 1–6, 2016.

[231] B. Soewito, F. L. Gaol, E. Simanjuntak, and F. E. Gunawan, “Smart mobile attendance system using voice recognition and fingerprint on smartphone,” in *Proceeding - 2016 International Seminar on Intelligent Technology and Its Application, ISITIA 2016: Recent Trends in Intelligent Computational Technologies for Sustainable Energy*, 2017, pp. 175–180.

[232] R. Cahyaningtiyas, R. Arianto, and E. Yosrita, “Fingerprint for automatic Door integrated with Absence and User Access,” in *2016 International Symposium on Electronics and Smart Devices, ISESD 2016*, 2017, pp. 26–29.

[233] D. J. Patterson, D. Fox, H. Kautz, and M. Philipose, "Fine-grained activity recognition by aggregating abstract object usage," *Proc. - Int. Symp. Wearable Comput. ISWC*, vol. 2005, pp. 44–51, 2005.

[234] G. A. Oguntala *et al.*, "SmartWall: Novel RFID-enabled Ambient Human Activity Recognition using Machine Learning for Unobtrusive Health Monitoring," *IEEE Access*, vol. PP, no. May, pp. 1–1, 2019.

[235] Y. Jiang *et al.*, "e-Textile embroidered wearable near-field communication RFID antennas," *IET Microwaves, Antennas Propag.*, vol. 13, no. 1, pp. 99–104, Jan. 2019.

[236] P. Lukowicz *et al.*, "Textile Building Blocks: Toward Simple, Modularized, and Standardized Smart Textile," in *Smart Textiles: Fundamentals, Design, and Interaction*, no. May, Springer International Publishing, 2017, pp. 303–331.

[237] M. Guibert *et al.*, "Washing reliability of painted, embroidered, and electro-textile wearable RFID tags," in *2017 Progress in Electromagnetics Research Symposium - Fall (PIERS - FALL)*, 2017, pp. 828–831.

[238] R. Kronberger, U. Dettmar, C. Hudasch, R. Lerche, M. Cremer, and A. Pervez, "A transmitter beamforming system for the localization of passive RFID tags," in *2016 IEEE Radio and Wireless Symposium (RWS)*, 2016, pp. 252–255.

[239] X. Jiang *et al.*, "Ultrasonic Fingerprint Sensor With Transmit Beamforming Based on a PMUT Array Bonded to CMOS Circuitry," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 3010, no. c, pp. 1–1, 2017.

[240] J. C. Kuo, J. T. Hoople, M. Abdelmejeed, M. Abdel-Moneum, and A. Lal, "64-Pixel solid state CMOS compatible ultrasonic fingerprint reader," *Proc. IEEE Int. Conf. Micro Electro Mech. Syst.*, pp. 9–12, 2017.

[241] Qualcomm, "Qualcomm Announces Advanced Fingerprint Scanning and Authentication Technology | Qualcomm," *Mobile World Congress Shanghai 2017*, 2017. [Online]. Available: https://www.qualcomm.com/news/releases/2017/06/28/qualcomm-announces-advanced-fingerprint-scanning-and-authentication. [Accessed: 19-Aug-2017].

[242] QualComm, "Fingerprint Sensors | Fingerprint Scanner Phone Security | Qualcomm." [Online]. Available: https://www.qualcomm.com/products/features/security/fingerprint-sensors. [Accessed: 19-Aug-2017].

[243] L. Chen, C. Nugent, and A. Al-Bashrawi, "Semantic data management for situation-aware assistance in ambient assisted living," *Proc. 11th Int. Conf. Inf. Integr. Web-based Appl. Serv. - iiWAS '09*, p. 298, 2009.

[244] Z. Jiang and S. Mao, "Energy Delay Tradeoff in Cloud Offloading for Multi-Core

Mobile Devices," *IEEE Access*, 2015.

[245] D. Martín, D. López-de-Ipiña, A. Alzua-Sorzabal, C. L. Lamsfus, and E. Torres-Manzanera, "A methodology and a web platform for the collaborative development of context-aware systems," *Sensors (Switzerland)*, 2013.

[246] B. Guo, D. Zhang, and M. Imai, "Toward a cooperative programming framework for context-aware applications," in *Personal and Ubiquitous Computing*, 2011.

[247] P. N. Borza, M. Romanca, and V. Delgado-Gomes, "Embedding patient remote monitoring and assistive facilities on home multimedia systems," in *2014 International Conference on Optimization of Electrical and Electronic Equipment, OPTIM 2014*, 2014.

[248] T. Kistel, O. Wendlandt, and R. Vandenhouten, "Using distributed feature detection for an assistive work system," in *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2014, pp. 1801–1802.

[249] A. De Paola, P. Ferraro, S. Gaglio, and G. Lo Re, "Autonomic behaviors in an Ambient Intelligence system," in *2014 IEEE Symposium on Computational Intelligence for Human-like Intelligence (CIHLI)*, 2014, pp. 1–8.

[250] A. Reichman and M. Zwiling, "The architecture of ambient assisted living system," in *2011 IEEE International Conference on Microwaves, Communications, Antennas and Electronic Systems (COMCAS 2011)*, 2011, pp. 1–4.

[251] PipelineDB, "PipelineDB - High-Performance Time-Series Aggregation for PostgreSQL." [Online]. Available: https://www.pipelinedb.com/. [Accessed: 03-Mar-2019].

[252] F. Holzschuher and R. Peinl, "Querying a graph database - Language selection and performance considerations," *J. Comput. Syst. Sci.*, vol. 82, no. 1, pp. 45–68, 2016.

[253] C. Küçükkeçeci and A. Yazıcı, "Big Data Model Simulation on a Graph Database for Surveillance in Wireless Multimedia Sensor Networks," *Big Data Res.*, vol. 11, pp. 33–43, 2018.

[254] M. Ayad, M. Taher, and A. Salem, "Real-time mobile cloud computing: A case study in face recognition," in *Proceedings - 2014 IEEE 28th International Conference on Advanced Information Networking and Applications Workshops, IEEE WAINA 2014*, 2014.

[255] S. Abolfazli, Z. Sanaei, E. Ahmed, A. Gani, and R. Buyya, "Cloud-based augmentation for mobile devices: Motivation, taxonomies, and open challenges," *IEEE Commun. Surv. Tutorials*, vol. 16, no. 1, pp. 337–368, 2014.

[256] Z. Li and K. H. Yap, “Context-aware Discriminative Vocabulary Tree Learning for mobile landmark recognition,” *Digit. Signal Process. A Rev. J.*, 2014.

[257] Shimmer Sensing, “Wearable Sensor Technology | Wireless IMU | ECG | EMG | GSR.” [Online]. Available: http://www.shimmersensing.com/. [Accessed: 26-Aug-2016].

[258] Q. Z. Sheng, X. Qiao, A. V. Vasilakos, C. Szabo, S. Bourne, and X. Xu, “Web services composition: A decade’s overview,” *Inf. Sci. (Ny).*, vol. 280, pp. 218–238, 2014.

[259] Jersey, “Chapter 15. Server-Sent Events (SSE) Support.” [Online]. Available: https://jersey.java.net/documentation/latest/sse.html#d0e11582. [Accessed: 10-Jun-2016].

[260] Apache, “Apache Jena - Apache Jena Fuseki.” [Online]. Available: https://jena.apache.org/documentation/fuseki2/. [Accessed: 08-Jun-2019].

[261] Libelium, “Waspmote Plug & Sense!,” *Waspmote Technical Guide*, 2013. [Online]. Available: http://www.libelium.com/products/plug-sense/. [Accessed: 01-May-2016].

[262] Robert Faludi, *Building Wireless Sensor Networks*, 1st ed. Sebastopol: O’Reilly Media, 2010.

[263] Tom Igoe, *Making Things Talk*, 2nd ed. Sebastopol: Maker Media, Inc, 2007.

[264] G. Meditskos, S. Dasiopoulou, V. Efstathiou, and I. Kompatsiaris, “SP-ACT: A hybrid framework for complex activity recognition combining OWL and SPARQL rules,” in *2013 IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops 2013*, 2013.

[265] Apache, “Apache Jena - Jena Ontology API.” [Online]. Available: https://jena.apache.org/documentation/ontology/. [Accessed: 08-Jun-2019].

[266] Google, “Google Products API.” [Online]. Available: https://developers.google.com/products/. [Accessed: 08-Jun-2019].

[267] W. Dai and V. Vyatkin, “A component-based design pattern for improving reusability of automation programs,” *IECON Proc. (Industrial Electron. Conf.*, pp. 4328–4333, 2013.

[268] X. Xu, Y. Tao, X. Wang, and X. Ding, “Research on Architecture of Smart Home Networks and Service Platform,” in *2014 5th International Conference on Digital Home*, 2014, pp. 232–236.

[269] R. K. Lomotey and R. Deters, “Sensor Data Propagation in Mobile Hosting Networks,” in *2015 IEEE Symposium on Service-Oriented System Engineering*, 2015, pp. 98–106.

[270] H. Knublauch, J. A. Hendler, and K. Idehen, “SPIN: SPARQL Inferencing Notation,” *W3C Member Submission*, 2011. [Online]. Available: http://spinrdf.org/. [Accessed: 07-

Oct-2016].

[271] W3C, “SWRL: A Semantic Web Rule Language Combining OWL and RuleML,” 2004. [Online]. Available: https://www.w3.org/Submission/SWRL/. [Accessed: 20-Aug-2016].

[272] P. Bota, J. Silva, D. Folgado, and H. Gamboa, “A Semi-Automatic Annotation Approach for Human Activity Recognition,” *Sensors (Basel).*, vol. 19, no. 3, pp. 1–23, 2019.